

# matematyka

## materiały metodyczne

2,71828182845904523846264338427427724769353354759457138177825186427427466291932003098621817413596920435729653429528265630758132328827943460763232382988675319525121901157326416792077021546891489348841675050447614606806326482016847741185374234544242710753077448920685517027618386062613313845832

**redakcja**

**Ryszard J. Pawlak**  
**Zofia Walczak**



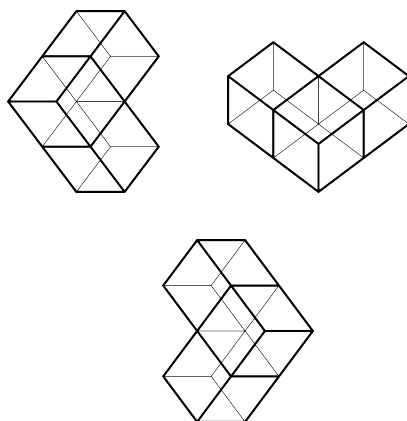
WYDAWNICTWO  
UNIwersytetu  
ŁÓDZKIEGO

X

---

# $\text{\LaTeX}$ – system składu dokumentów

---



Opracowanie

Aleksandra Hankus

Zofia Walczak



# ROZDZIAŁ 1

---

## Wiadomości wstępne

---

### Instalacja systemu

---

L<sup>A</sup>T<sub>E</sub>X jest formatem będącym częścią systemu T<sub>E</sub>X który instalujemy wybierając jedną z dwóch dystrybucji: MiK<sub>T</sub>E<sub>X</sub> lub T<sub>E</sub>XLive. Najbardziej znanymi dystrybucjami dla Microsoft Windows są T<sub>E</sub>XLive i MiK<sub>T</sub>E<sub>X</sub>. Obie dystrybucje możemy zainstalować bezpośrednio z internetu<sup>1</sup> lub z płyty DVD dystrybuowanej przez lokalne grupy użytkowników T<sub>E</sub>X-a, w Polsce to GUST (Grupa Użytkowników Systemu TeX).

Wszystkie dystrybucje T<sub>E</sub>X-a zawierają albo dokładną informację o instalacji albo graficzny instalator systemu.

### Wybieramy edytor

---

Do pracy z L<sup>A</sup>T<sub>E</sub>X-em możemy używać dowolnego edytora tekstu, jedynym warunkiem który musimy spełnić to zapisać nasz plik w kodzie ASCII i nadać mu nazwę w formacie `plik.tex`. W nazwie naszego pliku nie powinno być znaków diakrytycznych (np. *ę*, *ą*, *ń*), a także nie powinna zawierać spacji. Wprawdzie najnowsze dystrybucje T<sub>E</sub>XLive pozwalają na kompilowanie plików, w nazwach których zastosowano spacje lub litery ze znakami diakrytycznymi ale bezpieczniej jest z nich zrezygnować. W nazwie nie można także użyć znaków, które w T<sub>E</sub>X-u pełnią specjalną rolę. Są to znaki `#`, `$`, `%`, `&`, `\`, `^`, `_` oraz nawiasy klamrowe `{` i `}`.

---

<sup>1</sup>[www.tug.org/texlive/acquire-netinstall.html](http://www.tug.org/texlive/acquire-netinstall.html) i [miktex.org/2.9/setup](http://miktex.org/2.9/setup)

Wygodnie jest używać edytorów specjalnie dedykowanych do pracy z  $\text{\TeX}$ -em, które automatycznie zapisują naszą pracę w odpowiednim formacie. Większość z nich udostępnia szeroki wachlarz podpowiedzi składni środowisk latexowych oraz oznaczeń matematycznych. Takie edytory istnieją dla wszystkich systemów operacyjnych, można korzystać z edytorów darmowych, na licencji share-ware lub komercyjnych. Wśród edytorów darmowych można wymienić LEd, WinShell,  $\text{\TeX}$ works czy  $\text{\TeX}$ maker.

Po utworzeniu pliku dokumentu w wybranym edytorze aby uniknąć bałaganu, należy zapisać go w osobnym katalogu gdyż podczas kompilacji, oprócz pliku końcowego, tworzonych jest kilka plików pomocniczych. Zaleca się, aby nazwa takiego katalogu nie zawierała znaków diakrytycznych, gdyż w niektórych przypadkach, szczególnie w starszych wersjach systemu, może się zdarzyć, że nagle program przestanie działać.

## Co dalej z plikiem

---

Użytkownik  $\text{\LaTeX}$ -a ma do wyboru dwa kompilatory: `latex` i `pdflatex`. Od wybranego kompilatora zależy format pliku końcowego. W wyniku działania pierwszego otrzymujemy plik *nazwa.dvi* który możemy obejrzeć i wydrukować korzystając z programu `dviout` lub jego „starszego brata” `windvi`. Plik *nazwa.dvi* możemy następnie przetworzyć na *nazwa.ps* używając programu `dvips` i efekt naszej pracy obejrzeć za pomocą przeglądarki `GSview`. Jeśli skompilujemy nasz plik `pdflatex`-em otrzymamy plik *nazwa.pdf*, który możemy obejrzeć lub wydrukować używając dowolnej przeglądarki plików pdf, na przykład programu `Adobe Acrobat Reader`.

Podczas kompilacji naszego pliku powstaje kilka plików pomocniczych, między innymi plik *nazwa.log*. Znajdziemy w nim informacje o przebiegu kompilacji czyli np. o wczytanych pakietach oraz o błędach, jeśli takie wystąpiły. Poniżej przykład informacji zawartych w pliku *nazwa.log*.

**Przykład 1.** Poniżej pierwsze linie pliku *nazwa.log*.

```
This is pdfTeX, Version 3.1415926-2.5-1.40.14 (TeXLive2013/W32TeX)
```

```
(format=pdflatex2014.3.27)15NOV2014 14:54entering extended mode
\write18 enabled. %&-line parsing enabled.i tak dalej...
```

**Przykład 2.** Linie zawierające informację o przepelnieniu w wierszu (podane są numery linii tekstu).

```
[4] (c:/texlive/2011/texmf-dist/tex/latex/base/omscmr.fd)
LaTeX Warning: ‘!h’ float specifier changed to ‘!ht’.
[5]
Overfull \hbox (40.99998pt too wide) in paragraph at
lines 157--168 [] []
[6]
Overfull \hbox (7.39641pt too wide) in paragraph at lines
202--204
```

**Przykład 3.** I jeszcze informacja o błędzie w pliku powodującym zatrzymanie procesu kompilacji.

```
Package xcolor Warning: Incompatible color definition on
input line 409.
```

```
! LaTeX Error: Command \end{prob} invalid in math mode.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

```
...
```

```
1.431 \end{prob}
```

```
?
```

Większość edytorów wskazuje miejsce w którym wystąpił błąd umieszczając, po przerwaniu kompilacji, kursor w odpowiedniej linii tekstu i podając jej numer.

Jeżeli w naszym dokumencie znajdują się odsyłacze i spisy: treści, tabel, rysunków czy literatury, trzeba go skompilować co najmniej dwa razy.



## ROZDZIAŁ 2

---

# Tworzymy dokument

---

W dowolnym edytorze, na przykład T<sub>E</sub>Xworks otwieramy nowy dokument. Każdy dokument tekstowy który ma zostać przeczytany i przetworzony przez kompilator L<sup>A</sup>T<sub>E</sub>X-a, musi posiadać ściśle określoną strukturę, na którą składają się:

```
<prolog>
\documentclass[lista opcji]{nazwa klasy}
<preambuła>
\begin{document}
<Treść dokumentu>
\end{document}
```

W prologu możemy umieścić polecenia wywołujące programy zewnętrzne, które zostaną wykonane jeszcze przed uruchomieniem kompilatora. Najczęściej prolog pozostaje pusty, a dokument zaczyna się od wybrania klasy poleceniem `\documentclass`. Klasa dokumentu to pakiet zawierający definicje i deklaracje różnych parametrów dotyczących strony m. inn. szerokości i wysokości tekstu, rodzaju i wielkości użytych czcionek w tytułach itp.

L<sup>A</sup>T<sub>E</sub>X pozwala na używanie dowolnych klas, byleby tylko napisane były według ściśle określonego schematu. Czasopisma naukowe, szczególnie publikujące prace z zakresu nauk ścisłych, przygotowały własne klasy dokumentów i udostępniają je autorom artykułów na swoich stronach internetowych. Polski użytkownik L<sup>A</sup>T<sub>E</sub>X-a ma do dyspozycji standardowe klasy amerykańskie `article`, `book`, `report` i `letter` oraz ich polskie odpowied-



niki (uwzględniające nasze zwyczaje w piśmiennictwie naukowym) `mwart`, `mwbk` i `mwrep`.

Wymienione klasy mogą zostać wczytane z dodatkowymi parametrami umieszczonymi wewnątrz nawiasów kwadratowych w miejscu argumentu `lista opcji`. Mogą to na przykład być `a4paper`, `oneside` czy `12pt`. Regułą jest, że wszystkie parametry i opcje wymagane umieszczane są w nawiasach klamrowych, natomiast te nieobowiązkowe w nawiasach kwadratowych.

## Preambuła

---

Preambuła dokumentu  $\text{\LaTeX}$ -owego to ta jego część, która znajduje się pomiędzy poleceniami `\documentclass` i `\begin{document}`. Umieszczamy tam wszystkie deklaracje, które mają obowiązywać w całym dokumencie i mają wpływ na jego końcowy wygląd. W preambule wczytujemy również dodatkowe pakiety (czyli zestawy makrodefinicji) ponieważ podczas uruchamiania  $\text{\LaTeX}$ -a wczytywany jest tylko zestaw podstawowy, który umożliwia składanie prostych dokumentów.

Wszystkie pakiety wczytuje się poleceniem `\usepackage`. Niektóre pakiety posiadają zdefiniowane różne dodatkowe opcje, inne nie. Te, które nie mają żadnych dodatkowych opcji można wczytywać jednym poleceniem `\usepackage` umieszczając w nawiasach klamrowych po kolei ich nazwy i oddzielając je przecinkiem.

```
\usepackage[cp1250]{inputenc}
\usepackage{amssymb,amsmath,amsthm}
```

- Nie wstawiamy odstępów pomiędzy znakiem ukośnika wstecznego (`\`) a nazwą polecenia!
- Nie należy wstawiać odstępów pomiędzy `\begin` i nazwą środowiska.
- Niektóre nowe pakiety dopuszczają taki zapis ale generalna zasada brzmi: nie ma odstępów w tych miejscach.

Kolejność wczytywania pakietów nie jest obojętna. Część pakietów nie tylko wprowadza nowe polecenia ale także zmienia już istniejące (lub wczytane razem z innym pakietem) dlatego też należy dbać o to, by preambuła zawierała tylko te pakiety, które są niezbędne do złożenia bieżącego dokumentu.

Pakiet jest to zbiór makrodefinicji ułatwiających tworzenie dokumentu. Może go napisać każdy użytkownik L<sup>A</sup>T<sub>E</sub>X-a stosując zalecenia do tworzenia pakietów zamieszczone w literaturze. Nie wszystkie istniejące pakiety mogą być wykorzystywane przez początkującego użytkownika systemu. Te o szerokich zastosowaniach są dostarczane z kolejnymi implementacjami T<sub>E</sub>X-a.

Jeżeli użytkownik chce zdefiniować nowe polecenie czy środowisko lub zmienić (przedefiniować) już istniejące, może skorzystać z jednej z deklaracji `\newcommand`, `\renewcommand`, `\newenvironment` czy `\renewenvironment`. Jeżeli nowe polecenie ma obowiązywać w całym dokumencie, musi być umieszczone w preambule.

## Jak składać dokument w języku polskim?

---

Dokument w języku polskim możemy składać wpisując polskie litery w edytorze wszystkie po naciśnięciu prawego klawisza `Alt`, ale musimy poinformować kompilator L<sup>A</sup>T<sub>E</sub>X-a jakiego używamy systemu kodowania. Ostatnio najczęściej używanymi systemami kodowania są: utf8 i MS Windows (cp1250). W systemach MS Windows można też używać kodowania utf8, jeśli nasz edytor na to pozwala. O wybranym systemie kodowania informujemy kompilator poprzez wczytanie pakietu `inputenc` z właściwym dla danego kodowania argumentem. Preambuła naszego dokumentu powinna więc wyglądać następująco:

w systemie Windows:

```
\documentclass{article}
\usepackage[cp1250]{inputenc}
\usepackage{polski}
\begin{document}
```

lub

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\begin{document}
```

w systemie Linux:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\begin{document}
```

Wraz z pakietem `inputenc` wczytywany jest plik z tablicą konwersji właściwą dla strony kodowej naszego systemu operacyjnego. Zawiera on informacje jaki kod binarny odpowiada konkretnemu znakowi, czyli jaki znak pojawi się w dokumencie kiedy np. napiszemy go z prawym klawiszem `Alt`.

Do niedawna kody binarne znaków były opisywane na bajtach maksymalnie ośmiobitowych, co dawało możliwość opisania maksymalnie 256 różnych znaków. Powstały więc różne strony kodowe przydatnych do wprowadzania znaków z różnych języków. Problem stron kodowych został rozwiązany po wprowadzeniu kodowania Unicode. Najnowsze implementacje `TeXLive` i `MiKTeX` umożliwiają używanie kodowania `utf8` również w systemie MS Windows. W takim przypadku należy odpowiednio zmienić rodzaj kodowania w edytorze.

Wczytywany pakiet `polski` zmienia na polskie wszystkie napisy automatycznie generowane przez `LATEX`-a takie jak np. angielski tytuł Bibliography zamienia na Literatura, Chapter na Rozdział itd. Zmieniana też jest definicja makra `\today` które wstawia datę według reguł języka polskiego i z polskimi nazwami miesięcy. Wprowadza też kilka zmian w składzie matematyki, na przykład zmienia na polskie znaki *mniejsze-równe* ( $\leq$ ) czy definiuje nowe makra dla funkcji trygonometrycznych `tg` – `\tg`.

## Akcenty i znaki specjalne L<sup>A</sup>T<sub>E</sub>X-a

---

Przygotowując dokument należy pamiętać, że pewne znaki zostały zarezerwowane do specjalnych celów. Są to #, \$, %, &, \, ^, \_, {, }. Znak ukośnika wstecznego \ rozpoczyna każde polecenie T<sub>E</sub>X-a (i L<sup>A</sup>T<sub>E</sub>X-a), znak procentu (%) oznacza początek komentarza i działa do końca linii, znak dolara (\$) otwiera i zamyka środowisko matematyczne, znaki ^ i \_ oznaczają odpowiednio indeks górny i dolny zaś nawiasy {, } określają zakres działania danego polecenia. Znak & używany jest do wskazania przejścia do następnej komórki w wierszach tabeli. Używa się go także do wskazania miejsca wyrównania wzorów w środowisku wieloliniowym. Znaki ^, \_ i & mogą być użyte tylko w środowisku matematycznym. Znak # używany jest w definicji nowego polecenia lub środowiska do oznaczenia miejsca wstawienia parametru.

\$ = \ \$	& = \&	% = \%	# = \#
_ = \_	{ = \{	} = \}	
§ = \S	† = \dag	‡ = \ddag	
¶ = \P	© = \copyright	£ = \pounds	

Tabela 2.1: Znaki specjalne

Znaki specjalne pojawiające się w tekście należy poprzedzić znakiem ukośnika wstecznego \. Wyjątek stanowi sam znak \ – wywołujemy go poleceniem `\backslash`. Na klawiaturze nie znajdziemy także pewnych znaków przydatnych do składania różnorodnych tekstów. Dla nich zostały w T<sub>E</sub>X-u zdefiniowane specjalne polecenia (patrz tabela 2.1).

Czasami musimy w tekście umieścić wyrazy nie pochodzące z języka polskiego i zawierające litery z różnymi akcentami. W tabeli 2.2 przedstawiono instrukcje przydatne do uzyskania odpowiednich akcentów nad lub pod literami. Zastosowaliśmy je do liter e, o i u, ale można ich używać z dowolnymi literami.

\’o	ó	\’e	è	\^o	ô	\~o	õ
\=o	ō	\.e	è	\"o	ö	\ss	ß
\u{o}	ö	\v{e}	ě	\H{o}	ó	\c{o}	q
\oe	œ	\OE	Œ	\ae	æ	\AE	Æ
\aa	å	\AA	Å	\k{e}	ę		
\o	ø	\O	Ø	\l	ł	\L	Ł
\i	ı	\j	Ј	! ‘	ı	? ‘	ı

Tabela 2.2: Akcenty

Do składania całych dokumentów w językach obcych wykorzystujemy specjalne pakiety, które między innymi kontrolują wszystkie nazwy tytułów poszczególnych części dokumentu oraz przenoszenie wyrazów właściwe dla danego języka. Tak więc z instrukcji zamieszczonych w tabeli 2.2 korzystamy tylko do wpisywania do dokumentu pojedynczych obcych wyrazów, na przykład w spisie literatury.

## ROZDZIAŁ 3

---

# Jak sformatować tekst

---

Dokument  $\text{T}_{\text{E}}\text{X}$ -owy (i  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -owy) musi być przygotowany i zapisany w czystym kodzie ASCII, a jakiegokolwiek formatowanie tekstu, np. wszelką zmianę grubości, wielkości lub kształtu liter w tekście musimy specjalnie zaznaczyć.

Zwykły tekst wprowadzamy z klawiatury natomiast wszelkie wyróżnienia np. pogrubienie czy kursywę musimy poprzedzić odpowiednią instrukcją.

Poniżej zamieściliśmy instrukcje zmieniające odmianę pisma w trybie akapitowym. Należy pamiętać, by cały wyróżniony tekst był umieszczony w nawiasach klamrowych ograniczających działanie instrukcji.

## Rodzaj i grubość czcionki

---

`\textrm{tekst}` – pismo proste,  
`\textsl{tekst}` – *pismo proste pochylone*,  
`\textit{tekst}` – *kursywa* (italic),  
`\textbf{tekst}` – **pismo pogrubione** (bold),  
`\texttt{tekst}` – pismo imitujące pismo maszynowe,  
`\textsf{tekst}` – pismo bezszeryfowe.

W  $\text{T}_{\text{E}}\text{X}$ -u spotykamy się z rodzinami fontów takimi jak `\rmfamily`, `\sffamily` czy `\ttfamily`. W niektórych definicjach będziemy się odwoływać do rodziny fontu, a nie do polecenia używanego do napisania tekstu tym fontem.

```

\textit{Ten tekst napisany jest kursywą}
\textbf{Ten tekst napisany jest czcionką pogrubioną}
\textsl{Ten tekst napisany jest czcionką pochyłą}
\textup{Tekst napisany czcionką prostą}
\textsc{Tekst napisany kapitalikami}
\texttt{Ten tekst napisany jest czcionką imitującą
pismo maszynowe}

```

### Przykład 1.

*Ten tekst napisany jest kursywą*

**Ten tekst napisany jest czcionką pogrubioną**

*Ten tekst napisany jest czcionką pochyłą*

TEKST NAPISANY KAPITALIKAMI

Ten tekst napisany jest czcionką imitującą pismo maszynowe.

W formacie `plain` odmianę pisma zmienia się instrukcjami `\it`, `\bf`, `\tt` i `\sl`, ale działają one nieco inaczej niż te zdefiniowane w `LATEX`-u. Jeżeli do jakiegoś wyrazu zastosujemy instrukcję `\bf` i nie zamkniemy jej wraz z wyrazem w nawiasach klamrowych, to reszta naszego dokumentu będzie złożona czcionką pogrubioną.

Tekst możemy również wyróżnić poprzez pochylenie (domyślnie w klasie `article` jest to kursywa) używając jednej z następujących deklaracji `\em` lub równoważnej `\emph`. Obie te deklaracje różni tylko sposób w jaki powinny być wywoływane. Pierwsza działa dopóki nie zostanie użyta inna deklaracja zmiany fontu, dlatego należy cały tekst razem z deklaracją zamknąć w grupę za pomocą nawiasów klamrowych `{\em wyróżniony tekst}`, w drugiej natomiast zmieniany tekst umieszczamy w nawiasach klamrowych jako jej argument `\emph{wyróżniony tekst}`.

### Przykład 2.

Pierwsza deklaracja *pochylająca litery tekstu*.

Pierwsza deklaracja `{\em pochyłająca litery tekstu}`

Druga deklaracja *pochylająca litery tekstu*.

Druga deklaracja `\emph{pochylająca litery tekstu}`.

Podkreślenie dowolnego wyrazu lub ich grupy realizuje polecenie `\underline`.

### Przykład 3.

W niektórych przypadkach potrzebne jest wyróżnienie w tekście tylko jednego wyrazu przez podkreślenie natomiast wyróżnienie całego akapitu przez podkreślenie nie jest uważane za poprawne. Należy pamiętać, że w tekście wyróżnionym podkreśleniem L<sup>A</sup>T<sub>E</sub>X nie przenosi wyrazów, mogą więc powstawać wiersze nadmiarowe.

## Wielkość czcionki

---

Domyślną wielkością czcionką dla dokumentu L<sup>A</sup>T<sub>E</sub>X-owego jest czcionka 10-cio punktowa prosta. Dostępne są też wielkości 11 i 12 punktów (1 cal = 72,27 pt, 1 cm = ok. 28,45 pt).

L<sup>A</sup>T<sub>E</sub>X ma zdefiniowane deklaracje różnych wielkości czcionek od bardzo małej (`\tiny`) do bardzo dużej (`\Huge`). Przykładowe wielkości można znaleźć w tabeli 3.1. Używając tych deklaracji należy pamiętać o grupowaniu powiększanego tekstu w nawiasach klamrowych.

### Przykład 4.

Zastosujemy najpierw **powiększenie** używając polecenia `{\huge powiększenie}`. Powiększyliśmy tylko jeden wyraz dzięki zgrupowaniu w nawiasach klamrowych polecenia i wyrazu powiększanego. Teraz „zapomnimy” o grupowaniu i **zobaczymy co się stanie**

W drugim przypadku powiększony został cały tekst a nie tylko jeden wyraz następujący po instrukcji zwiększającej czcionkę. Aby zakończyć działanie instrukcji powiększającej i powrócić do wielkości domyślnej dla naszego dokumentu należy wywołać instrukcję `\normalsize`.



Stopień pisma tekstu głównego	10pt	11pt	12pt	
<code>\tiny</code>	5pt	6pt	6pt	najmniejsza
<code>\scriptsize</code>	7pt	8pt	8pt	bardzo mała
<code>\footnotesize</code>	8pt	9pt	10pt	nieco większa
<code>\small</code>	9pt	10pt	11pt	mała
<code>\normalsize</code>	10pt	11pt	12pt	rozmiar normalny
<code>\large</code>	12pt	12pt	14pt	duża
<code>\Large</code>	14pt	14pt	17pt	większa
<code>\LARGE</code>	17pt	17pt	20pt	jeszcze większa
<code>\huge</code>	20pt	20pt	25pt	jeszcze większa
<code>\Huge</code>	25pt	25pt	25pt	największa

Tabela 3.1: Predefiniowane wielkości czcionek

Jeżeli musimy złożyć część tekstu naszego dokumentu czcionką większą niż dostępne w L<sup>A</sup>T<sub>E</sub>X-u, nie należy stosować instrukcji z tabeli 3.1.

Wygodnie jest skorzystać z możliwości wczytywanego domyślnie pakietu `anyfontsize`. Dostarcza on instrukcje `\fontsize{}{}` i `\selectfont`. Użyjemy ich w następnym przykładzie. **Przykład 5.**

Ten dokument jest składany czcionką domyślną (10-punktową). Te wyrazy są powiększone do 14 punktów, te dwa do 18 punktów a ten wyraz jest napisany czcionką 24-punktową.

```
domyślną (10-punktową).\fontsize{14}{14}\selectfont Te wyrazy}
{\fontsize{18}{18}\selectfont te dwa} do 18~punktów
{\fontsize{24}{24}\selectfont ten wyraz} jest napisany
```

Argument umieszczony w pierwszej klamrze instrukcji `\fontsize` to rozmiar czcionki (w punktach) a argument w drugiej parze kontroluje odległość pomiędzy wierszami (jeśli taka potrzeba zaistnieje). Instrukcja `\selectfont` powoduje rozpoczęcie składania tekstu wybranym fontem. Cały powiększa-

ny tekst razem z omówionymi instrukcjami należy zgrupować w nawiasach klamrowych. Takie zgrupowanie zagwarantuje nam zastosowanie instrukcji tylko do konkretnego tekstu. Zakończenie działania omawianych instrukcji można też uzyskać instrukcją `\normalsize`, która informuje kompilator o powrocie do wielkości domyślnej fontu.

Do powiększania czcionek w całym dokumencie wygodnie jest użyć klas `extsizes` lub zastosować powiększenie dostępne po wczytaniu pakietu `geometry`.

## Odstępy

---

T<sub>E</sub>X składa wiersze i akapity inaczej niż standardowe edytory. Kompilator „czyta” cały akapit tekstu i dzieli go na wiersze, więc rozpoczynając składanie dokumentu należy pamiętać, że

- bez względu na to ile umieścimy odstępów (spacji) pomiędzy dwoma wyrazami, w dokumencie pojawi się tylko jeden, czyli jeden odstęp poziomy znaczy tyle samo co wiele, tylko pierwszy jest brany pod uwagę,
- wszystkie odstępny na początku linii są ignorowane,
- odstęp kończący komendę jest usuwany,
- przejście do nowej linii (CR) jest traktowane jak pojedynczy odstęp,
- każdy akapit kończymy pozostawiając pustą linię lub umieszczając na jego końcu instrukcję `\par`,
- dwie, trzy lub więcej wolnych linii w tekście znaczy tyle samo co jedna.

Ponieważ powyższe własności T<sub>E</sub>X-a (L<sup>A</sup>T<sub>E</sub>X-a) wykluczają możliwość uzyskania w dokumencie odstępu innego niż domyślny międzywyrazowy czy międzywierszowy, zdefiniowane zostały osobne polecenia umożliwiające wstawienie dowolnego odstępu poziomego lub pionowego.

## Wcięcie akapitowe

Rozpoczyna każdy akapit tekstu. Jego wielkość regulowana jest poleceniem `\parindent`. Jeżeli przed rozpoczęciem nowego akapitu umieścimy deklarację `\setlength{\parindent}{3cm}` to następny i kolejne akapity będą się rozpoczynać trzy-centymetrowym wcięciem. Jeśli kilka akapitów tekstu ma mieć wcięcie inne niż pozostałe, należy zgrupować je umieszczając w nawiasach klamrowych razem z deklaracją `\setlength{\parindent}{3cm}`.

Wcięcie akapitowe likwiduje instrukcja `\noindent` a instrukcja `\indent` wstawia wcięcie na początku wiersza, który nie rozpoczyna akapitu. W klasie `article` początkowy akapit punktu (section) rozpoczyna się bez wcięcia akapitowego natomiast w klasie `mwart` z wcięciem.

## Odstęp poziomy

Poziomy odstęp który będzie zmieniał się razem ze zmianą wielkości czcionki uzyskamy za pomocą poleceń `\quad` i `\qquad` lub przez umieszczenie pomiędzy wyrazami znaku ukośnika wstecznego (w-tył-ciach lub ang. backslash) i pustej spacji, czyli `\_`. Pierwsze z nich (`\quad`) dodaje do pojedynczego odstęp równy szerokości litery m w bieżącym foncie, drugie (`\qquad`) dwa razy więcej, a trzecie (`\_`) standardowy odstęp międzywyrazowy w bieżącym foncie.

Dodatkowe, bardzo małe odstępy można też uzyskać umieszczając bezpośrednio po znaku sterującym dwukropek, przecinek, średnik lub wykrzyknik `\,`, `\;`, `\:`, `\!`. Wszystkie one działają w środowisku matematycznym i są bardzo pomocne przy pisaniu matematycznych wzorów (instrukcję `\,` można używać również w trybie tekstowym).

## Odstęp poziomy dowolnej długości

Wstawiając pomiędzy dwa wyrazy polecenie `\hspace{odległość}` lub `\hspace*{odległość}` uzyskamy poziomy odstęp o długości `odległość`. Odległość ta musi być podana razem z jednostką np. `\hspace{2cm}`. Polecenie bez gwiazdki wstawione na początku wiersza jest ignorowane, natomiast

Polecenie `\hfill`, będące skróceniem polecenia `\hspace{fill}`, spowoduje wypełnienie spacjami przestrzeni między wyrazami i „rozepchnięcie” ich do lewego i prawego marginesu.

Początek wiersza                      środek                      i koniec.

Była sobie ..... żabka mała  
która mamy ..... nie słuchała  
na ..... spacerek ..... wychodziła  
wszystkim ..... żabkom ..... się dziwiła.

Harcerze \_\_\_\_\_ wesoło śpiewali.

W przykładzie 8 odległość między wyrazami *Harcerze* i *wesoło śpiewali* została podzielona na trzy części i środkowa została wypełniona kreską.

## Odstęp pionowy

Gdy umieszczony pomiędzy akapitami jest wykonywany tylko wtedy, kiedy poprzedzający go akapit został zakończony pustą linią lub poleceniem `\par`. W przeciwnym wypadku polecenie zrobienia odstępu zostanie wykonane dopiero po zakończeniu kolejnego akapitu lub po zakończeniu wiersza.

Zwiększenie odstępu pomiędzy dwoma akapitami uzyskamy poleceniem `\vspace{odległość}` (lub `\vspace*{odległość}`). Parametr `odległość` musi być liczbą (dodatnią lub ujemną) z mianem (np. `1cm`, `3ex`, itp). Ujemny parametr wstawia ujemny odstęp (czyli zmniejsza odstęp istniejący). Polecenie z gwiazdką wymusza wstawienie odstępu tam, gdzie normalnie L<sup>A</sup>T<sub>E</sub>X nie dopuszcza takiej możliwości, a więc np. na początku strony.

Odległość pomiędzy akapitami ustalana jest automatycznie wraz z wyborem wielkości fontu. Jest to wielkość „rozciągliwa” wyrażana w jednostkach `ex` (`1ex` jest równy wysokości małej litery `x` w bieżącym foncie). Odległość pomiędzy akapitami reguluje polecenie `\parskip`, zmienia się ją poleceniem `\setlength{\parskip}{odległość}`, gdzie jako `odległość` należy podać również w jakich granicach może być ona rozciągnięta, np. warunek `\setlength{\parskip}{2ex plus 0.5ex minus 0.5ex}` dopuszcza zwiększenie lub zmniejszenie odległości między akapitami o `0.5ex`.

Jeśli zmienimy odstęp pomiędzy akapitami wewnątrz dokumentu, to będzie on taki do końca dokumentu. Aby zmiana dotyczyła całego dokumentu należy polecenie zmieniające `\parskip` umieścić w preambule, jeśli tylko wybranych akapitów, należy te akapity zgrupować w nawiasach klamrowych razem z poleceniem zmieniającym odstęp.

Aby zwiększyć odległość pomiędzy dwoma akapitami możemy także użyć poleceń `\smallskip`, `\medskip` i `\bigskip` wymienionych tutaj od najmniejszego odstępu do największego.

## Zmiana odstępów międzywierszowych

Aby powiększyć odstępy pomiędzy wierszami wystarczy w preambule umieścić instrukcję `\linespread{liczba}`, gdzie argument `liczba` jest większy

od 1 gdy chcemy odstępy powiększyć, a mniejszy od 1 gdy chcemy je zmniejszyć. Domyślnie argument ten jest równy 1, co odpowiada pojedynczemu odstępowi międzywierszowemu. Jego wielkość jest dobierana w zależności od wielkości użytego fontu.

Ten dokument jest złożony z odstępem pomiędzy wierszami równym 1.3 odstępu pojedynczego, a więc argument `liczba` jest równy 1.3.

## Pozycjonowanie tekstu na stronie

---

### Środowisko `center`

---

Zazwyczaj każdy wiersz dokumentu jest wyrównany do obu marginesów ale możemy też rozmieścić go inaczej. Do umieszczania całego akapitu centralnie na stronie należy skorzystać ze środowiska `center` w następujący sposób:

```
\begin{center}
```

```
Akapit tekstu
```

```
\end{center}
```

Cały tekst zostanie automatycznie podzielony na wiersze krótsze niż szerokość tekstu i każdy z nich zostanie wyśrodkowany. Jeśli zależy nam na podzieleniu tekstu na linie o konkretnej długości lub zawartości, wówczas każdą z nich musimy zakończyć instrukcją `\\`.

```
\begin{center}
```

```
pierwsza linia tekstu \\
```

```
....\\
```

```
k-ta linia tekstu
```

```
\end{center}
```

### Przykład 9.

Tekst wyśrodkowany bez znaków łamania linii.

Cały tekst zostanie automatycznie podzielony na wiersze krótsze niż szerokość tekstu i każdy z nich zostanie wyśrodkowany. Jeśli zależy nam na podzieleniu tekstu na linie o konkretnej długości lub zawartości, wówczas każdą z nich musimy zakończyć instrukcją `\\`.

### Przykład 10.

Cały tekst zostanie automatycznie  
podzielony na wiersze krótsze niż szerokość tekstu  
i każdy z nich zostanie wyśrodkowany.

Jeśli zależy nam na podzieleniu tekstu na linie  
o konkretnej długości lub zawartości, wówczas każdą z nich  
musimy zakończyć instrukcją `\\`.

Do wyśrodkowania tekstu jednego krótkiego wiersza można użyć polecenia `\centerline{tekst}`.

### Środowiska `flushleft` i `flushright`

---

Do wyrównania wierszy akapitu do jednego z marginesów służą środowiska `flushleft` (wyrównanie do lewego marginesu) lub środowiska `flushright` (wyrównanie do prawego marginesu). W obu przypadkach możemy samodzielnie podzielić wiersze lub zostawić tę pracę kompilatorowi.

Podobny efekt otrzymamy, gdy formatowany tekst umieścimy w środowisku `raggedleft` (odpowiednik `flushright`) lub `raggedright` (odpowiednik `flushleft`).

Jeśli użyjemy środowiska `flushleft` i nie zastosujemy ręcznego łamania linii, to otrzymamy skład tekstu w tzw. chorągiewkę.

Dla wyrównania tekstu do prawego lub lewego marginesu można także użyć jednego z poleceń

```
\flushleft{tekst}, \flushright{tekst}  
\raggedleft{tekst}, \raggedright{tekst}.
```

Różnica pomiędzy środowiskami a poleceniami o tych samych nazwach polega na tym, że w przypadku tych ostatnich każda linia koniecznie musi być zakończona znakiem łamania (`\`). Należy też pamiętać, że w żadnym z tych środowisk L<sup>A</sup>T<sub>E</sub>X nie przenosi wyrazów.

### Przykład 11.

Poniżej przykład zastosowania środowisk `flushright` i `flushleft` wraz z ręcznym łamaniem wierszy tekstu.

```
\begin{flushleft}
pierwszy wiersz \\
drugi wiersz \\
....\\
wiersz n-ty
\end{flushleft}
```

W Canmore Kowalczyk ściagała się na swoim ulubionym dystansie,  
w ulubionym stylu i na trasie, która wyjątkowo jej pasuje.

W klasyfikacji generalnej PŚ Polka jest już druga i wiele wskazuje na to,  
że z Canmore wyjedzie w żółtej kamizelce liderki PŚ.  
Kolejny start w sobotę. Transmisja w Eurosporcie.

W Canmore Kowalczyk ściagała się na swoim ulubionym dystansie,  
w ulubionym stylu i na trasie, która wyjątkowo jej pasuje.

W klasyfikacji generalnej PŚ Polka jest już druga i wiele wskazuje na to,  
że z Canmore wyjedzie w żółtej kamizelce liderki PŚ.

Kolejny start w sobotę. Transmisja w Eurosporcie.

### Przykład 12.

Polecenie `\raggedright` możemy zastosować do dowolnego tekstu tak, by każdy wyraz zaczynał się w nowym wierszu, wyliczając na przykład kolejne miesiące.

Styczeń (środa)

Luty (wtorek)



Marzec (poniedziałek)

Kwiecień (niedziela)

...

### Przykład 13.

Poniżej tekst złożony „w chorągiewkę” w środowisku `flushleft` (to znaczy bez ręcznego łamania wierszy).

```
\begin{flushleft}
treść akapitu
\end{flushleft}
```

Z dawnych tradycji drukarstwa polskiego wiele dowodów kunsztu literniczego znajdujemy w zachowanych starych drukach. W muzeach i bibliotekach zachowane zostały wzorce rodzimej twórczości. Próby stworzenia rodzimych warsztatów projektowania liternictwa drukarskiego podejmowało dawniej rzemiosło artystyczne.

Już w okresie starego państwa egipskiego w ramach egipskiego systemu pisma ukształtował się swoisty system złożony z 24 znaków do oznaczania spółgłosek.<sup>1</sup>

## Środowiska `quote` i `quotation`

---

Do innego wyróżnienia tekstu służą środowiska `quote` i `quotation`. W każdym z nich tekst jest sformatowany w bloku węższym niż szerokość tekstu i jednakowo odległym od obu marginesów.

```
\begin{quote} tekst \end{quote}
\begin{quotation} tekst \end{quotation}
```

Różnica między tymi dwiema formami polega na tym, że w środowisku `quote` akapity nie zaczynają się od wcięcia, zaś w środowisku `quotation`

---

<sup>1</sup>Wyjątek z książki *Pismo i styl*, Tibor Szántó

tekst każdego akapitu rozpoczyna się wcięciem, odległość pomiędzy akapitami jest równa odległości między wierszami. W obu przypadkach dodawany jest dodatkowy odstęp przed i za wyróżnionym tekstem.

**Przykład 14.**

Ten sam tekst, pochodzący z książki *Pismo i styl*, Tibora Szántó, umieścimy najpierw w środowisku `quote` a potem w `quotation`. Podzielimy go także na akapity.

Cztery tysiące lat przed naszą erą ukształtowały się w Egipcie dwa rodzaje pisma. Pierwszy, bardziej płynny rodzaj pisma, ...

Już w okresie starego państwa egipskiego w ramach egipskiego systemu pisma ukształtował się swoisty system złożony z 24 znaków do oznaczania spółgłosek.

Pismo hieroglificzne znane jest na ogół jako pismo obrazkowe.

---

Cztery tysiące lat przed naszą erą ukształtowały się w Egipcie dwa rodzaje pisma. Pierwszy, bardziej płynny rodzaj ...

Już w okresie starego państwa egipskiego w ramach egipskiego systemu pisma ukształtował się swoisty system złożony z 24 znaków do oznaczania spółgłosek.

Pismo hieroglificzne znane jest na ogół jako pismo obrazkowe.



## ROZDZIAŁ 4

---

# Projektujemy stronę

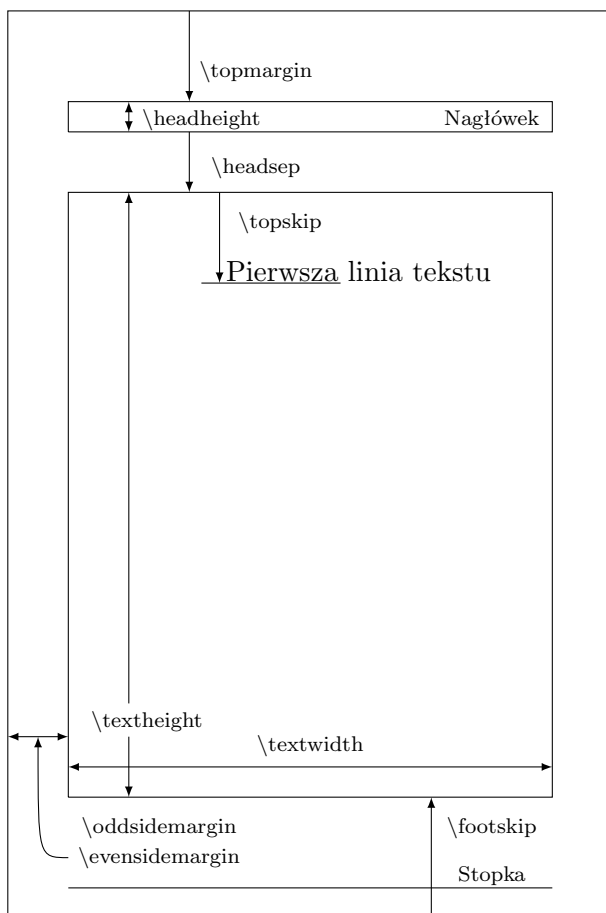
---

Planując złożenie dokumentu należy zaplanować jego wygląd, a więc zdecydować:

- O objętości dokumentu (liczba stron);
- O podziale dokumentu na części (rozdziały, punkty itd.);
- Czy dokument będzie składany jednostronnie czy dwustronnie;
- Jaki będzie rozmiar papieru na którym będzie drukowany dokument;
- Ustalić wielkości marginesów, wielkości i rodzaj czcionki oraz wielkości interlinii;
- Określić wygląd stopki i nagłówka strony oraz sposób numerowania stron;
- O składzie jedno- czy dwukolumnowym.

Objętość dokumentu implikuje wybór klasy  $\text{\LaTeX}$ -owej której użyjemy do jego składania. Podstawowe klasy dokumentów to **article** – do składania krótkich artykułów naukowych, **report** – do składania dłuższych rozpraw naukowych oraz **book** – do składania książek. W każdej z tych klas zdefiniowane są podstawowe cechy strony i ustawione jako domyślne.

Domyślne ustawienia parametrów strony możemy zmieniać używając poleceń zdefiniowanych w pakietach **geometry** (patrz rozdział 8) oraz **fancyhdr**, który pozwala na własne sformatowanie stopek i nagłówków. Oba te pakiety dostarczane są z każdą dystrybucją  $\text{\TeX}$ -a.



Rysunek 4.1: Parametry układu graficznego strony

Domyślne wartości parametrów przedstawionych na rysunku 4.1 można też zmienić bez korzystania z pakietu `geometry`. Należy wtedy w preambule umieścić polecenie `\setlength{nazwa}{wielkość}`.

## Strona

---

Styl strony to inaczej jej wygląd. Możemy go zadeklarować umieszczając w preambule polecenie `\pagestyle{nazwa stylu}`. W  $\text{\LaTeX}$ -u zdefiniowano następujące style:

- `plain` – nagłówek (`head`) strony jest pusty, stopka (`foot`) zawiera wycentrowany numer strony. Ten styl przyjmowany jest jako domyślny, jeżeli żaden inny nie jest zdefiniowany w `preamble`.
- `empty` – nagłówek i stopka są puste, nie drukują się także numery stron.
- `headings` – nagłówek zawiera numer strony oraz nazwę i tytuł rozdziału (paragrafu, zależnie od klasy dokumentu). Na stronie z tytułem nagłówek jest pusty. Stopka jest zawsze pusta.
- `myheadings` – numer strony tak samo jak w stylu `headings` znajduje się w nagłówku, a tekst trzeba podać samodzielnie stosując polecenia `\markright{tekst}` (w dokumencie drukowanym jednostronnie) i `\markboth{lewy tekst}{prawy tekst}` (w dokumencie drukowanym dwustronnie). Stopka, tak jak w stylu `headings` jest pusta.

Aby strona dokumentu nie miała numeru, należy gdziekolwiek w tekście na tej stronie umieścić polecenie `\thispagestyle{empty}`. Następna strona będzie miała numer kolejny (poprzednia będzie policzona, lecz numer nie będzie drukowany).

## Numery stron

---

Dokumenty składane w podstawowych klasach (`article`, `mwart`, `report`, `mwrep`, `book`, `mbwk`) strony są numerowane cyframi arabskimi. Polecenie `\pagenumbering{styl numerowania}` pozwala zmienić styl numerowania na jeden z następujących:

- `arabic` – liczby arabskie (domyślnie);
- `roman` – liczby rzymskie małe;
- `Roman` – liczby rzymskie duże;
- `alph` – małe litery alfabetu łacińskiego;

- Alph – duże litery alfabetu łacińskiego.

Numerując strony literami alfabetu łacińskiego (małymi lub dużymi) musimy pamiętać, że jest ich tylko 26 i tylko tyle stron może mieć nasz dokument.

Numerowaniem stron steruje licznik o nazwie `page` i w każdym miejscu dokumentu możemy nadać mu inną wartość niż kolejna za pomocą instrukcji `\setcounter{page}{numer}`. Stronie na której pojawi się ta instrukcja, zostanie nadana liczbą „numer” a następne otrzymają numery kolejne.

Nie zawsze musimy numerować strony od pierwszej. Aby rozpocząć numerowanie stron naszego dokumentu od strony drugiej wystarczy w preambule napisać: `\setcounter{page}{2}`.

## Pakiet fancyhdr

---

Makra dostarczone wraz z pakietem `fancyhdr` (wczytuje się go w preambule poleceniem `\usepackage{}`) pozwalają na łatwe przededefiniowanie nagłówków i stopków w naszym dokumencie. Możemy więc:

- Zdefiniować trzyczęściowe nagłówki i stopki;
- Wstawiać różnego rodzaju linie oddzielające nagłówki i stopki od tekstu głównego;
- Nagłówki i stopki szersze niż tekst główny;
- Wieloliniowe nagłówki i stopki;
- Inne nagłówki i stopki na stronach parzystych i nieparzystych;
- Inne nagłówki i stopki na stronie z tytułem rozdziału i na stronach zawierających rysunki lub tabele;
- Kontrolować rodzaj i rozmiar fontu w nagłówkach i stopkach.

Rysunek poniżej pokazuje wszystkie elementy strony które możemy modyfikować.

lewy nagłówek	środkowy nagłówek	prawy nagłówek
treść dokumentu		
lewa stopka	środkowa stopka	prawa stopka

## Jak to zrobić?

Najpierw należy poleceniem `\usepackage{}` wczytać pakiet `fancyhdr`, następnie umieścić deklarację stylu stron poleceniem `\pagestyle{fancy}`. Po wczytaniu pakietu będziemy mogli definiować poszczególne elementy nagłówków i stopek korzystając z następujących instrukcji:

- `\lhead` — lewy nagłówek,
- `\chead` — środkowy nagłówek,
- `\rhead` — prawy nagłówek,
- `\lfoot` — lewa stopka,
- `\cfoot` — środkowa stopka,
- `\rfoot` — prawa stopka.

Powyższe instrukcje działają tylko w dokumencie składanym jednostronnie. W składzie dwustronnym poszczególne części nagłówka (stopki) na stronach parzystych i nieparzystych różnią się, należy więc rozróżnić je przy projektowaniu nagłówków i stopek.

Do projektowania nagłówków i stopek używamy wtedy poleceń

```
\fancyhead[miejsce]{treść}
\fancyfoot[miejsce]{treść}
```

gdzie parametr „miejsce” może przyjmować następujące wartości:



E	Strona parzysta (even)
O	Strona nieparzysta (odd)
L	Lewa strona
C	Środkowa strona
R	Prawa strona
H	Nagłówek
F	Stopka

Poniżej przykłady zastosowania omówionych sposobów projektowania nagłówków i stopek.

### Przykład 1.

Strony nieparzyste

Absolwenci Wydziału Prawa 2012		
Lista absolwentów		
Od: Dziekan	Do: Rektor UW	3

Strony parzyste

Absolwenci Wydziału Prawa 2012		
Lista absolwentów		
4	Od: Dziekan	Do: Rektor UW

Nagłówek i stopkę w przykładzie 1 zdefiniowaliśmy następująco:

```
\fancyhead{} % wszystkie nagłówki puste
\fancyhead[R0,LE]{ Absolwenci Wydziału Prawa 2012}
\fancyfoot{} % wszystkie stopki puste
\fancyfoot[LE,R0]{\thepage}
```

```
\fancyfoot[L0,CE]{Od: Dziekan}
\fancyfoot[CO,RE]{Do: Rektor UW}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

Za linie oddzielające tekst nagłówka i stopki od treści dokumentu odpowiadają polecenia `\headrule` i `\footrule`. Domyślnie są to linie ciągłe, o ustalonej grubości, ale możemy je przededefiniować tak, aby otrzymać np. linię złożoną z kropek. Należy wtedy przededefiniować polecenie `\headrule` i `\footrule` następująco:

```
\renewcommand{%
\headrule}{\vbox to 0pt{\hbox to \headwidth{\dotfill}\vss}%
}%
```

Szerokość tych linii możemy też zmienić poleceniem `\renewcommand`.

```
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.6pt}
```

Pakiet `fancyhdr` pozwala też na zmianę szerokości nagłówka i stopki. Domyślnie oba te elementy mają szerokość tekstu. Możemy je zwęzić lub poszerzyć używając poleceń:

```
\fancyheadoffset[miejsce]{długość}
\fancyfootoffset[miejsce]{długość}
\fancyhfoffset[miejsce]{długość}
```

Są one podobne do poleceń `\fancyfoot` i `\fancyhf` z tą jednak różnicą, że nie akceptują opcji C. Pozostałe akceptowalne miejsca to E, O, L i R. Jeśli parametr „długość” będzie dodatni, to otrzymamy nagłówek (stopkę) szerszy, jeśli ujemny to węższy.

Jeżeli wczytamy w preambule dokumentu pakiet `lastpage` będziemy mogli zdefiniować numery stron postaci „ $n$  z  $m$ ”. Należy wtedy przededefiniować część stopki (nagłówka) w której ten numer ma się znajdować. Poniżej przykład dla numeru w stopce, w środkowej jej części.

```
\cfoot{\thepage\ z \pageref{LastPage}}
```

Tak zdefiniowana numeracja stron pojawia się w nagłówkach lub w stopkach, ale nie w odsyłaczach, w spisach treści i indeksach.

W dokumentacjach technicznych pojawia się też numerowanie stron w obrębie rozdziału, czyli postaci „2-10”, gdzie pierwszy numer oznacza numer rozdziału, a drugi jest numerem strony tego rozdziału. Aby uzyskać takie numerowanie w dokumencie należy użyć pakietu **chappg**. Pakiet ten resetuje również licznik stron po zakończeniu rozdziału i w nowym rozdziale rozpoczyna numerowanie od pierwszej strony.

Aby styl numerowania „2-10” pojawił się również w spisie treści, odsyłaczach czy w indeksie, musimy przededefiniować odpowiednio `\thepage`.

Opis wielu innych poleceń pakietu **fancyhdr** można znaleźć w dokumentacji dostarczanej wraz z dystrybucją **T<sub>E</sub>XLive** lub w katalogach The Comprehensive **T<sub>E</sub>X**Archive Network ([ctan.org](http://ctan.org)).

## Tekst w kilku kolumnach

---

**L<sup>A</sup>T<sub>E</sub>X** domyślnie składa tekst w jednej kolumnie. Jeżeli w deklaracji klasy dokumentu umieścimy opcję `twocolumn`, wówczas cały dokument zostanie złożony w dwóch kolumnach. Jednak skład tekstu jaki otrzymamy nie będzie doskonały. Dużo lepiej złożony dokument otrzymamy, gdy skorzystamy z możliwości pakietu **multicol** autorstwa Franka Mittelbacha. Pakiet wczytuje się w preambule poleceniem `\usepackage{}`. Możemy użyć zdefiniowanego tam środowiska `multicols`, które pozwala nie tylko złożyć cały dokument w więcej niż jedna kolumnach, ale także przełączać się z jednej na dwie (i więcej) kolumn na jednej stronie. Pakiet ten jest dostępny w każdej dystrybucji **T<sub>E</sub>X**-a.

Tekst przeznaczony do złożenia w kilku kolumnach umieszczamy w środowisku `multicols` podając jako jego argument ilość kolumn.

```
\begin{multicols}{2}
```

```
\dots Wszystkie te misterne działania
```

`\end{multicols}`

Poniżej przykład, który zilustruje działanie pakietu `multicol`. Przy składaniu tekstu w dwóch i więcej kolumnach należy pamiętać, że długie wyrazy mogą nie mieścić się w pojedynczym wierszu. Składanie tekstu w dwóch lub więcej kolumnach może spowodować niechciane „dziury” pomiędzy wyrazami. Zawsze w takim przypadku należy zastanowić się czy nie lepiej będzie złożyć nasz tekst w jednej, szerszej kolumnie.

## Przykład 2.

... Wszystkie te misterne działania służą im do zdobycia właściwego męża, w odpowiednim czasie. Japońskie małżeństwa, w większości przypadków są aranżowane i stanowią swoisty układ między dwoma rodzinami, które wyraziły zgodę na wzajemne powinowactwo.

Musiało minąć już ze cztery godziny, od czasu gdy kobieta wyszła do pracy. Akurat druga grupa odchodziła z kosztami piasku w kierunku trójkołowego samochodu. Przysłuchiwał się uważnie i gdy stwierdził, że mężczyźni nie wracają, wstał cicho i włożył ubranie. Kobieta zabrała lampę, musiał więc zrobić wszystko po omacku. Kobieta była całkowicie pochłonięta pracą. ...<sup>1</sup>

---

<sup>1</sup>Fragment książki Kobo Abe „Kobieta z wydm”



## ROZDZIAŁ 5

---

# Składamy dokument

---

### Wprowadzanie tekstu

---

Podczas składania tekstu  $\text{\LaTeX}$  znajduje się w trybie akapitowym. Kompilator najpierw „czyta” cały akapit, potem dzieli go na wiersze, to samo dzieje się z następnymi akapitami aż w pamięci będzie wystarczająca ilość tekstu do zapelnienia strony. Wtedy następuje łamanie strony czyli podział tekstu na wiersze i akapity. Dzięki takiej organizacji udaje się uniknąć niechcianych odstępów między wyrazami i pomiędzy akapitami.

Jeżeli w trakcie czytania treści dokumentu  $\text{\LaTeX}$  natrafi na tekst matematyczny (zawarty np. pomiędzy znakami dolara) wtedy automatycznie przełącza się na tryb matematyczny.

Na tryb LR (left-right)  $\text{\LaTeX}$  przełącza się zawsze wtedy, gdy napotka jedno z poleceń tworzących pudełko np. `\parbox{ }{ }` lub `\mbox{ }`. W trybie LR tekst nie jest formatowany i sami musimy zadbać o łamanie wierszy .

### Tytuł, autor i data

---

Formatowanie logiczne dokumentu operuje takimi pojęciami, jak tytuł rozdziału, tytuł punktu, tabela, tytuł tabeli, odsyłacz, przypis, notka na marginesie, spisy treści, tabel, rysunków, literatury itp. i ma zdecydowaną przewagę nad wizualnym, zwiększa wygodę pracy i jej wydajność.

Akapit jest najmniejszą ale również najważniejszą częścią dokumentu. W L<sup>A</sup>T<sub>E</sub>X-u każdy z wymienionych elementów struktury powstaje poprzez zadeklarowanie odpowiedniej instrukcji lub polecenia.

Rozpoczynając pracę z dokumentem należy określić jego strukturę logiczną. Do tej struktury możemy także zaliczyć tytuł dokumentu, jego autora oraz (jeśli to konieczne) datę powstania.

Tytuł, nazwisko autora oraz data powstania dokumentu tworzą zbiór informacji określany jako „tytuł”. Wszystkie informacje umieszcza się zwykle w preambule dokumentu czyli przed poleceniem `\begin{document}`. Nazwisko autora jest argumentem polecenia `\author{nazwisko}`, tytuł dokumentu jest argumentem polecenia `\title{tytuł}`. W klasie `article` i `mwart` bieżąca data jest dodawana automatycznie. Jeśli nie chcemy daty w ogóle, powinniśmy umieścić polecenie `\date{}`, zaś jeśli chcemy wstawić datę inną od bieżącej wtedy piszemy na przykład `\date{2.02.2014}`. Aby wszystkie podane informacje znalazły się w dokumencie należy bezpośrednio po `\begin{document}` umieścić instrukcję `\maketitle`.

## Spisy treści, tabel, rysunków i literatury

---

*Spis treści* generowany jest instrukcją `\tableofcontents` którą wstawiamy w miejscu w którym chcemy go umieścić. Spis treści może być umieszczony na początku dokumentu lub na jego końcu. W krótkich dokumentach spis treści umieszcza się zazwyczaj na końcu. Aby cały spis został utworzony należy dwukrotnie uruchomić kompilator.

Jeśli w dokumencie występują tabele i rysunki to ich spisy są generowane instrukcjami `\listoftables` i `\listoffigures` odpowiednio. Aby spisy tabel i rysunków pojawiły się w spisie treści, należy odpowiednie instrukcje umieścić przed instrukcją generującą spis. W tym przypadku dokument także musi być przetworzony dwukrotnie.

Każda praca naukowa powinna zawierać spis literatury. W zależności od klasy dokumentu spis literatury umieszczany jest na osobnych stronach

lub bezpośrednio po zakończeniu artykułu. W klasie `article` spis literatury umieszczany jest bezpośrednio po tekście artykułu i zatytułowany jest *Literatura*, w klasie `report` na kolejnej stronie, a w klasie `book` na następnej stronie nieparzystej. W obu tych klasach ma tytuł *Bibliografia*.

Spis literatury umieszczamy w środowisku `thebibliography`, a kolejne jego pozycje rozpoczynamy instrukcją `\bibitem`:

```
\begin{thebibliography}{szer-etykiety}
\bibitem[etykieta]{klucz} autor, tytuł itd.
.....
\end{thebibliography}
```

Bez opcjonalnego argumentu `etykieta` L<sup>A</sup>T<sub>E</sub>X numeruje kolejne pozycje liczbami w nawiasach kwadratowych, inne oznaczenia (np. literowe) wprowadzamy ręcznie zamiast argumentu `etykieta`. L<sup>A</sup>T<sub>E</sub>X nie sortuje pozycji literatury, o właściwą kolejność należy zadbać samodzielnie.

Spis literatury będzie poprawnie sformatowany, gdy jako argument `szer-etykiety` umieścimy najdłuższą kombinację liter (lub cyfr) występujących jako etykiety kolejnych pozycji.

Jeżeli kolejne pozycje są numerowane, to jako argument `szer-etykiety` wstawiamy cyfrę 9, gdy pozycji literatury jest mniej niż 10, liczbę 99, gdy pozycji literatury jest mniej niż 100 i tak dalej.

Obowiązkowy argument `klucz` może składać się z dowolnej kombinacji liter, cyfr i znaków z wyjątkiem znaku przecinka i znaków specjalnych. Argument `klucz` umieszczamy w instrukcji `\cite{klucz}`, gdy chcemy w tekście dokumentu odwołać się do konkretnej pozycji spisu literatury.

Poniżej przykład numerowanego (sposób domyślny) spisu literatury oraz spisu literatury, którego kolejne pozycje są oznaczone kombinacją liter.

```
\begin{thebibliography}{9}
\bibitem{} Robert Chwałowski, Typografia typowej książki,
Helion 2002.
\bibitem{} Andrzej Tomaszewski, Architektura książki
{\small dla wydawców, redaktorów, poligrafów, grafików,
```



autorów, księgoznawców i~bibliofilów}, COBRPP 2011.

\end{thebibliography}

\begin{thebibliography}{MMM}

\bibitem[RCh]{} Robert Chwałowski, Typografia typowej książki, Helion 2002.

\bibitem[AT]{} Andrzej Tomaszewski, Architektura książki  
{\small dla wydawców, redaktorów, poligrafów, grafików,  
autorów, księgoznawców i~bibliofilów}, COBRPP 2011.

\end{thebibliography}

W przypadku spisu literatury należy również pamiętać o dwukrotnym przetworzeniu dokumentu latex-em.

## Pudełka

---

L<sup>A</sup>T<sub>E</sub>X oferuje użytkownikom możliwość umieszczenia akapitu w pudełku, które możemy dowolnie przesuwac na stronie. Rozróżniamy trzy rodzaje pudełek: typu LR (left-right) w którym zawartość jest umieszczana horyzontalnie od lewej do prawej, pudełko akapitu, w którym zawartość umieszczana jest pionowo wiersz po wierszu oraz pudełko używane zazwyczaj do rysowania pionowych lub poziomych linii.

Pudełka typu LR tworzą następujące polecenia: `\mbox`, `\makebox`, `\fbox` i `\framebox`.

```
\mbox{tekst}      \makebox[szerokość][pozycja]{tekst}
\fbox{tekst}      \framebox[szerokość][pozycja]{tekst}
```

Parametr `szerokość` musi zawierać liczbę wraz z jednostką, natomiast jako parametr `pozycja` mogą być podane następujące litery:

- l - wyrównywanie tekstu do lewego marginesu,
- r - wyrównywanie tekstu do prawego marginesu,
- s - rozstrzelenie tekstu po całym pudełku.

Polecenie	Wynik
<code>\mbox{Treść}</code>	To jest Treść
<code>\makebox[2cm]{Treść}</code>	To jest      Treść
<code>\fbox{Treść}</code>	To jest <span style="border: 1px solid black; padding: 2px;">Treść</span>
<code>\framebox[4cm][l]{Treść}</code>	To jest <span style="border: 1px solid black; display: inline-block; width: 100px; height: 1.2em; vertical-align: middle;">Treść</span>
<code>\framebox[4cm][r]{Treść}</code>	To jest <span style="border: 1px solid black; display: inline-block; width: 100px; height: 1.2em; vertical-align: middle; text-align: right;">Treść</span>
<code>\framebox[4cm][s]{Treść}</code>	To jest <span style="border: 1px solid black; display: inline-block; width: 100px; height: 1.2em; vertical-align: middle; text-align: center;">Treść      w      pudełku</span>
<code>\framebox[2mm]{centered}</code>	centered <span style="border: 1px solid black; display: inline-block; width: 10px; height: 1.2em; vertical-align: middle;"></span>
<code>\makebox[0pt][l]{/\$}</code>	To jest \$

Jak wiadomo T<sub>E</sub>X umieszcza litery, wyrazy i akapity w pudełkach, możemy więc tymi pudełkami „poruszać”. I tak do podniesienia lub opuszczenia wyrazu w stosunku do linii bazowej korzystamy z polecenia `\raisebox`

Linia bazowa `\raisebox{1ex}{wyżej}` a teraz

`\raisebox{-1ex}{niżej}` i znowu na linii podstawowej.

Linia podstawowa <sup>wyżej</sup> i teraz <sub>niżej</sub> i znowu na linii podstawowej.

Do umieszczania całych akapitów w pudełku, którego położeniem w dokumencie można sterować, możemy zastosować polecenie `\parbox` lub użyć środowiska `minipage`. Zdefiniowane one są następująco:

```
\parbox[pozycja]{szerokość}{tekst}
\begin{minipage}[pozycja]{szerokość}
tekst
\end{minipage}
```

gdzie parametr `pozycja` może przybierać wartości:

b – równanie do dolnego brzegu pudełka,

t – równanie do górnego brzegu pudełka,

a parametr `szerokość` musi być liczbą z mianem, na przykład 10 cm, 6 in.

Następne przykłady pokażą zastosowanie polecenia `\parbox`.

### Przykład 1.

To jest przykład  
pudełka o szerokości 3cm wyrównanego w pionie do środka linii bazowej (domyślny parametr c)

BIEŻĄCA LINIA

To jest przykład pudełka o szerokości 5cm wyrównanego w pionie do środka linii bazowej (domyślny parametr c)

```
\parbox{3cm}{To jest przykład pudełka o szerokości 3cm
wyrównanego w pionie do \dots}
\hfill BIEŻĄCA LINIA \hfill
\parbox{5cm}{To jest przykład pudełka o szerokości 5cm
wyrównanego w pionie do \dots}
```

Pudełka `minipage` i `parbox` można też ze sobą łączyć.

Należy jeszcze wspomnieć o różnicach dzielących `minipage` i `parbox`. Środowisko `minipage` zachowuje się tak jak cała strona dokumentu, można w nim umieszczać wszystkie te polecenia i instrukcje, które umieszcza się na stronie. Wyjątek stanowi przypis wywołany poleceniem `\footnote`. Jego treść znajdzie się u dołu ministrony a nie u dołu bieżącej strony tekstu.

Z uwagi na to, że `\parbox` jest poleceniem, którego argumentem jest tekst umieszczany w pudełku, nie możemy w tym pudełku używać żadnych innych instrukcji.

## Środowiska typu lista

---

W `LATEX`-u mamy trzy środowiska tworzące listy: `itemize`, `enumerate` oraz `description` oraz środowisko `list`, w którym możemy utworzyć swoją listę o własnych parametrach. W każdym ze zdefiniowanych środowisk kolejny punkt listy rozpoczyna się określoną etykietą. W przypadku listy numerowanej jest to kolejny numer (najczęściej napisany cyframi arabskimi), w przypadku listy nienumerowanej – etykieta zależy od wybranej klasy dokumentu. Poniżej przykłady środowisk `itemize`, `enumerate` i `description`

## Środowisko description

---

```
\begin{description}
\item[Etykieta] pierwszy punkt listy
\item[Etykieta] drugi punkt listy
.....
\end{description}
```

**Etykieta** pierwszy punkt listy

**Etykieta** drugi punkt listy

## Środowisko itemize i enumerate

---

W środowisku `itemize` tekst kolejnych punktów w klasie `article` poprzedzony jest grubą czarną kropką jako etykietą zaś w klasie `mwart` myślnikiem. Kropka taka jest wywoływana w T<sub>E</sub>X-u poleceniem `$\bullet$` (w trybie matematycznym). Poniżej przykład listy wypunktowanej.

### Przykład 2.

- Treść pierwszego punktu listy w klasie `mwart`.
- Treść drugiego punktu listy w klasie `mwart`.

```
\begin{itemize}
\item Treść pierwszego punktu listy w~klasie \texttt{mwart}.
\item Treść drugiego punktu listy w~klasie \texttt{mwart}.
\end{itemize}
```

### Przykład 3.

- Treść pierwszego punktu listy w klasie `article`.
- Treść drugiego punktu listy w klasie `article`.

```

\begin{itemize}
\item Treść pierwszego punktu listy w~klasie \texttt{article}.
\item Treść drugiego punktu listy w~klasie \texttt{article}.
\end{itemize}

```

W środowisku numerującym (`enumerate`) treść kolejnych punktów poprzedzona jest kolejnym numerem jako etykietą. \*

```

\begin{enumerate}
\item To jest punkt pierwszy.
\item To jest punkt drugi.
\item To jest punkt trzeci.
\end{enumerate}

```

Środowiska `enumerate` możemy zagnieżdżać aż do czwartego poziomu, to samo ze środowiskami `itemize`. Możemy także mieszać je ze sobą. Następny przykład pokazuje system numerowania czterech poziomów w środowisku `enumerate` w klasie `mwart`. Każda klasa dokumentu może mieć zdefiniowany inny system numeracji.

#### Przykład 4.

1. To jest pierwszy poziom listy `enumerate`
  - (a) A to drugi poziom
    - i. I już trzeci poziom listy
      - A. I wreszcie czwarty poziom
    - ii. To znowu trzeci
  - (b) A to już drugi
2. I wracamy na pierwszy poziom

## Środowisko `list`

---

Użytkownik L<sup>A</sup>T<sub>E</sub>X-a może zbudować własne środowisko typu lista definiując, odpowiednio do swoich potrzeb, poszczególne argumenty środowiska `list`. Środowisko to wywołuje się tak samo jak `enumerate` czy `itemize` a wartości parametrów możemy zadeklarować bezpośrednio po wywołaniu środowiska.

```
\begin{list}{etykieta}{lista_deklaracji}
\item tekst
\item tekst
\end{list}
```

Parametr `etykieta` ustala domyślną etykietę rozpoczynającą kolejne punkty listy. Wszystkie będą mieć taką samą etykietę ale w dowolnym punkcie listy możemy ją zmienić (na chwilę) poprzez umieszczenie innego znaku po poleceniu `\item` jak poniżej:

```
\begin{list}{\#\#}{}
\item Znak ustalony w definicji listy.
\item[\&] A ten punkt ma zmieniony znak rozpoczynający go.
\item I znowu ustalony w definicji znak.
\end{list}
```

Efekt widzimy w przykładzie 5.

### Przykład 5.

Zdefiniujemy listę z ustaloną etykietą dla wszystkich punktów a następnie w jednym punkcie zamienimy ją na inną.

# To jest punkt rozpoczynający się znakiem ustalonym w definicji listy.

# To też jest punkt rozpoczynający się znakiem ustalonym w definicji listy.

& A ten punkt ma zmienioną etykietę.

# I znowu ustalony w definicji znak.

W definicji powyższej listy druga para nawiasów klamrowych pozostała pusta co oznacza, że użyte zostały domyślne ustawienia odstępów poziomych i pionowych listy. Zajmiemy się teraz parametrem `lista_deklaracji` czyli spisem geometrycznych parametrów formatujących naszą listę, takich jak szerokość etykiety, odstęp pomiędzy kolejnymi punktami listy itd.

`\topsep` dodaje dodatkowy odstęp pionowy przed i po liście.

`\partopsep` dodaje dodatkowy odstęp od góry jeśli środowisko poprzedza pusta linia.

`\itemsep` dodaje dodatkowy odstęp pomiędzy punktami listy (dodany do domyślnego odstępu między wierszami).

`\parsep` dodaje dodatkowy odstęp pomiędzy akapitami wewnątrz punktu listy.

`\leftmargin` ustala poziomy odstęp pomiędzy lewym marginesem a listą (musi być nieujemny).

`\rightmargin` ustala poziomy odstęp pomiędzy prawym marginesem a listą (musi być nieujemny).

`\listparindent` umożliwia zwiększenie (zmniejszenie) wcięcia akapitu po etykiecie.

`\itemindent` ustala wcięcie pierwszej linii tekstu kolejnego punktu listy (może być ujemne).

`\labelsep` ustala poziomy odstęp pomiędzy końcem pudełka zawierającego etykietę a tekstem.

`\labelwidth` domyślna szerokość pudełka zarezerwowanego na etykietę; jeżeli etykieta jest większa niż zarezerwowane miejsce – zajmuje część miejsca w pierwszym wierszu punktu listy,

`\makelabel{label}` tworzy etykietę drukowaną poleceniem `\item`.

## Kilka przykładów list

Poniżej przedstawimy kilka przykładów list z różnymi parametrami. Będą one dobierane tak, by od razu można było zobaczyć co zostało zmienione. Na początek standardowa lista wypunktowana umieszczona w środowisku `itemize`. Każdy przykład będzie zawarty pomiędzy liniami demonstrującymi aktualną szerokość tekstu.

---

Szerokość tekstu

---

- Pierwsza lista jest umieszczona w środowisku `itemize` o domyślnych parametrach geometrycznych.
- Drugi punkt tej samej listy umożliwia zaobserwowanie domyślnych odstępów pomiędzy kolejnymi jej punktami.

---

Szerokość tekstu

---



---

Szerokość tekstu

---

- Parametr `\leftmargin` ma teraz wartość 2 cm, reszta ustawień domyślna.

---

Szerokość tekstu

---

⇒ Tutaj zmienimy jeszcze parametr `\labelsep` i nadamy mu wartość 1 cm, pierwszy parametr listy pozostał taki jak poprzednio, reszta ustawień domyślna.

⇒ Drugi punkt listy.

---

Szerokość tekstu

---

Teraz parametrowi `\labelsep` nadaliśmy wartość ujemną (−0.5 cm), parametrowi `\labelwidth` wartość 1 cm, pierwszy parametr listy pozostawiliśmy taki jak poprzednio, a reszta ustawień jest domyślna.

Drugim punktem listy. Okazuje się, że w tym przypadku etykieta została wydrukowana na tekście.



Parametry środowiska `list` pozwalają na skonstruowanie listy odpowiadającej bardzo różnym potrzebom użytkownika. Zajmowaliśmy się dotychczas tylko listami wypunktowanymi, ale za pomocą środowiska `list` można też utworzyć listę numerowaną. Wystarczy poleceniem `\newcounter` utworzyć licznik, a następnie wstawić go w miejsce deklaracji etykiety. Poniżej podajemy przykład najprostszej listy tego rodzaju:

I. Ta lista jest numerowana liczbami rzymskimi.

II. Utworzyliśmy nowy numer o nazwie *numer*.

```
\newcounter{numer}
\begin{list}{\Roman{numer}. }{\usecounter{numer}}
\item Ta lista jest numerowana liczbami rzymskimi.
\item Utworzyliśmy nowy numer o nazwie \textit{numer}.
\end{list}
```

## ROZDZIAŁ 6

---

# Grafika

---

## Pakiet `graphicx`

---

Możliwości graficzne L<sup>A</sup>T<sub>E</sub>X-a możemy rozszerzyć wczytując pakiet **graphicx**. Jest on rozszerzeniem wcześniejszego pakietu **graphics**, pakiety te różnią się nieznacznie, głównie formatem argumentów opcjonalnych w poleceniach. W typowej dystrybucji systemu T<sub>E</sub>X oba te pakiety są instalowane domyślnie.

### Podstawowe opcje pakietu

---

**draft** Powoduje wydrukowanie dokumentu bez wstawionej grafiki, zaznacza jednak wymiary pudełka, w które ten obrazek ma być wstawiony.

**final** Przeciwnie do opcji poprzedniej. Używane często do nadpisania opcji [draft] zadeklarowanej w opcjach klasy wczytywanych wraz z poleceniem

```
\documentclass.
```

**demo** Zamiast obrazka wstawianego instrukcją `\includegraphics` rysuje prostokąt o wymiarach 150pt na 100pt, nawet gdy rozmiar obrazka jest inny.

Najczęściej używane opcje to **draft** i **final**. Pakiet umożliwia wstawianie grafiki do dokumentu oraz obracanie i skalowanie obrazów i tekstu.

## Obracanie

---

Obrót jest realizowany za pomocą instrukcji `\rotate`, którą stosujemy następująco:

```
\rotatebox{kąt}{text}
\rotatebox{lista argumentów}{kąt}{text}
```

Pierwsza instrukcja pochodzi z pakietu `graphics`, wstawia obiekt (tekst) w pudełko, na przykład `\mbox` i obraca o kąt podany jako argument. Jako środek obrotu wybierany jest w tym przypadku lewy dolny wierzchołek pudełka.

### Przykład 1.

Wybieramy jako *środek* obrotu w tym przypadku lewy *dolny* wierzchołek *pudełka* i powracamy do linii bazowej.

```
Wybieramy jako \rotatebox{30}{środek} obrotu (...)
lewy \rotatebox{45}{dolny} wierzchołek
\rotatebox{60}{pudełka} i powracamy ... .
```

Jeśli za środek obrotu chcemy przyjąć inny punkt powinniśmy użyć drugiego polecenia, wybierając właściwy argument z „listy”. Są to

- `origin = <etykieta>`
- `x = <rozmiar>`
- `y = <rozmiar>`
- `units = <numer>`.

Można podać obydwie współrzędne (x,y) środka obrotu albo w argumencie 'origin' jako etykietę podać jedną lub dwie litery z następujących `lrcbtB`, gdzie B oznacza linię bazową. Poniżej przykład obrotu o 180° bez przesunięcia środka obrotu (kropki wyznaczają położenie linii bazowej):

...Przykładowy tekst...

i po przesunięciu w centrum linii bazowej w prawo do góry

... Przykładowy tekst ...  
... Przykładowy tekst ...

Można też zmieniać domyślny kierunek obrotu (przeciwnie do ruchu wskazówek zegara) korzystając z argumentu 'units' na przykład [units = -360] dla obrotu w kierunku ruchu zegara i [units= 6.283185] dla obrotu o kąt podany w radianach.

## Skalowanie

---

Pakiet `graphicx` umożliwia też skalowanie wstawianych obiektów. Składnia polecenia jest następująca:

```
\scalebox {h-skala}[v-skala]{tekst}
```

Można podać tylko poziomy argument skalowania a pionowy zostanie dobrany proporcjonalnie.

### Przykład skalowania

Instrukcja `\reflectbox{tekst}` pozwala otrzymać tekst w odbiciu lustrzanym. Jest to skrócona forma polecenia `\scalebox{-1}[1]{tekst}`.

ntskt iicidbo emsrtatw tzej oT

Instrukcja `\resizebox` umożliwia skalowanie obiektu do zadanej wielkości np.

```
\resizebox{h-length}{v-length}{text}
```

```
\resizebox{1in}{\height}{Przykładowy tekst}
```

Przykładowy tekst

Tekst ma szerokość 1 cala i wysokość tekstu bieżącego.

```
\resizebox{2in}{!}{Przykładowy tekst}
```

## Przykładowy tekst

Tu wysokość została wyliczona stosownie do szerokości.

```
\resizebox{4cm}{1.5cm}{Przykładowy tekst}
```

# Przykładowy tekst

Tutaj tekst ma wysokość 1.5 cm a szerokość 4 cm.

## Wstawianie plików graficznych do dokumentu

---

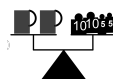
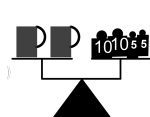
Polecenie wstawiające plik graficzny ma postać:

```
\includegraphics[lista arg]{plik}
```

Poniżej lista wybranych argumentów polecenia.

- `angle` – kąt obrotu grafiki,
- `width=<liczba>` – szerokość,
- `height=<liczba>` – wysokość,
- `origin` – środek obrotu - patrz `\rotatebox`,
- `scale=<liczba>` – liczba określająca skalę powiększenia.

Poniżej przykład ilustrujący użycie parametrów `width` i `height`.



```
\includegraphics{waga.png}
```

```
\includegraphics  
[width=1.5cm]{waga.png}
```



```
\includegraphics[height=1cm]  
{waga.png}
```

```
\includegraphics[height=2cm,  
width=1cm]{waga.png}
```

Pliki graficzne mogą znajdować się w dowolnych katalogach, w poleceniu `\graphicspath{lista katalogów}` zamiast listy katalogów można napisać np. `\graphicspath{{./eps/}{./tiff/}}`.

Następne polecenie ułatwia systemowi zadanie kiedy nie podamy typu pliku graficznego `\DeclareGraphicsExtensions{ext-list}`

Lista rozszerzeń (ext-list) to nazwy oddzielone od siebie przecinkiem. Można także podać ścieżkę dostępu do pliku bezpośrednio przed jego nazwą:

```
\includegraphics{D:/Rysunki/kolo.png}
```



## ROZDZIAŁ 7

---

# Wybrane symbole matematyczne

---

### Akcenty w trybie matematycznym

---

$\hat{o}$ <code>\hat{o}</code>	$\tilde{o}$ <code>\tilde{o}</code>	$\widehat{o}$ <code>\widehat{o}</code>
$\widetilde{o}$ <code>\widetilde{o}</code>	$\check{o}$ <code>\check{o}</code>	$\breve{o}$ <code>\breve{o}</code>
$\dot{o}$ <code>\dot{o}</code>	$\ddot{o}$ <code>\ddot{o}</code>	$\bar{o}$ <code>\bar{o}</code>
$\vec{o}$ <code>\vec{o}</code>	$\grave{o}$ <code>\grave{o}</code>	$\acute{o}$ <code>\acute{o}</code>

### Podstawowe symbole relacji

---

$<$ <code>&lt;</code>	$>$ <code>&gt;</code>	$\leq$ <code>\le</code>	$\geq$ <code>\ge</code>
$\ll$ <code>\ll</code>	$\gg$ <code>\gg</code>	$\doteq$ <code>\doteq</code>	$\simeq$ <code>\simeq</code>
$=$ <code>=</code>	$\equiv$ <code>\equiv</code>	$\sim$ <code>\sim</code>	$\approx$ <code>\approx</code>
$\subset$ <code>\subset</code>	$\supset$ <code>\supset</code>	$\in$ <code>\in</code>	$\ni$ <code>\ni</code>
$ $ <code>\mid</code>	$\parallel$ <code>\parallel</code>	$\notin$ <code>\notin</code>	$\neq$ <code>\neq</code>
$\neq$ <code>\not=</code>	$\nparallel$ <code>\not\parallel</code>	$\not>$ <code>\not&gt;</code>	$\not<$ <code>\not&lt;</code>

### Symbole niektórych operacji dwuargumentowych

---

$+$ <code>+</code>	$-$ <code>-</code>	$\pm$ <code>\pm</code>	$\mp$ <code>\mp</code>
--------------------	--------------------	------------------------	------------------------



$\cdot$ <code>\cdot</code>	$\div$ <code>\div</code>	$\times$ <code>\times</code>	$\backslash$ <code>\setminus</code>
$\star$ <code>\star</code>	$*$ <code>\ast</code>	$\circ$ <code>\circ</code>	$\bullet$ <code>\bullet</code>
$\cup$ <code>\cup</code>	$\cap$ <code>\cap</code>	$\vee$ <code>\vee</code>	$\wedge$ <code>\wedge</code>
$\oplus$ <code>\oplus</code>	$\ominus$ <code>\ominus</code>	$\odot$ <code>\odot</code>	$\otimes$ <code>\otimes</code>

## Symbole zmiennej wielkości

---

$\sum$ <code>\sum</code>	$\bigcap$ <code>\bigcap</code>	$\bigcup$ <code>\bigcup</code>
$\bigsqcup$ <code>\bigsqcup</code>	$\prod$ <code>\prod</code>	$\coprod$ <code>\coprod</code>
$\bigvee$ <code>\bigvee</code>	$\bigwedge$ <code>\bigwedge</code>	$\bigoplus$ <code>\bigoplus</code>
$\int$ <code>\int</code>	$\iint$ <code>\iint</code>	$\oint$ <code>\oint</code>
$\odot$ <code>\bigodot</code>	$\biguplus$ <code>\biguplus</code>	$\bigotimes$ <code>\bigotimes</code>

## Niektóre strzałki

---

$\leftarrow$ <code>\leftarrow</code>	$\rightarrow$ <code>\rightarrow</code> lub <code>\to</code>
$\longleftarrow$ <code>\longleftarrow</code>	$\longrightarrow$ <code>\longrightarrow</code>
$\Lleftarrow$ <code>\Lleftarrow</code>	$\Rrightarrow$ <code>\Rrightarrow</code>
$\Longleftarrow$ <code>\Longleftarrow</code>	$\Longrightarrow$ <code>\Longrightarrow</code>
$\leftrightarrow$ <code>\leftrightarrow</code>	$\longleftrightarrow$ <code>\longleftrightarrow</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\Longleftrightarrow$ <code>\Longleftrightarrow</code>
$\mapsto$ <code>\mapsto</code>	$\longmapsto$ <code>\longmapsto</code>
$\hookrightarrow$ <code>\hookrightarrow</code>	$\hookleftarrow$ <code>\hookleftarrow</code>
$\Uparrow$ <code>\Uparrow</code>	$\Downarrow$ <code>\Downarrow</code>
$\uparrow$ <code>\uparrow</code>	$\downarrow$ <code>\downarrow</code>
$\Updownarrow$ <code>\Updownarrow</code>	$\Downarrow$ <code>\Downarrow</code>
$\updownarrow$ <code>\updownarrow</code>	$\Updownarrow$ <code>\Updownarrow</code>
$\nearrow$ <code>\nearrow</code>	$\searrow$ <code>\searrow</code>
$\nwarrow$ <code>\nwarrow</code>	$\swarrow$ <code>\swarrow</code>

## Litery greckie

---

$\alpha$ \alpha	$\theta$ \theta	$\varnothing$ \o	$\upsilon$ \upsilon
$\beta$ \beta	$\vartheta$ \vartheta	$\pi$ \pi	$\phi$ \phi
$\gamma$ \gamma	$\iota$ \iota	$\varpi$ \varpi	$\varphi$ \varphi
$\delta$ \delta	$\kappa$ \kappa	$\rho$ \rho	$\chi$ \chi
$\epsilon$ \epsilon	$\lambda$ \lambda	$\varrho$ \varrho	$\psi$ \psi
$\varepsilon$ \varepsilon	$\mu$ \mu	$\sigma$ \sigma	$\omega$ \omega
$\zeta$ \zeta	$\nu$ \nu	$\varsigma$ \varsigma	$\varkappa$ \varkappa
$\eta$ \eta	$\xi$ \xi	$\tau$ \tau	$\digamma$ \digamma
$\aleph$ \aleph	$\beth$ \beth	$\daleth$ \daleth	$\gimel$ \gimel
$\Gamma$ \Gamma	$\Lambda$ \Lambda	$\Sigma$ \Sigma	$\Psi$ \Psi
$\Delta$ \Delta	$\Xi$ \Xi	$\Upsilon$ \Upsilon	$\Omega$ \Omega
$\Theta$ \Theta	$\Pi$ \Pi	$\Phi$ \Phi	

## Litery – różne style (tylko w środowisku matematycznym)

---

Kaligraficzne:  $\mathcal{A}$  \mathcal{A} ... itd.

$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}, \mathcal{M}, \mathcal{N}, \mathcal{O}, \mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{Y}, \mathcal{Z}$

Mathbb:  $\mathbb{A}$  \mathbb{A} ... itd.

$\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E}, \mathbb{F}, \mathbb{G}, \mathbb{H}, \mathbb{I}, \mathbb{J}, \mathbb{K}, \mathbb{L}, \mathbb{M}, \mathbb{N}, \mathbb{O}, \mathbb{P}, \mathbb{Q}, \mathbb{R}, \mathbb{S}, \mathbb{T}, \mathbb{U}, \mathbb{V}, \mathbb{W}, \mathbb{Y}, \mathbb{Z}$

Mathfrak:  $\mathfrak{A}$  \mathfrak{A} ... itd.

$\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}, \mathfrak{E}, \mathfrak{F}, \mathfrak{G}, \mathfrak{H}, \mathfrak{I}, \mathfrak{J}, \mathfrak{K}, \mathfrak{L}, \mathfrak{M}, \mathfrak{N}, \mathfrak{O}, \mathfrak{P}, \mathfrak{Q}, \mathfrak{R}, \mathfrak{S}, \mathfrak{T}, \mathfrak{U}, \mathfrak{V}, \mathfrak{W}, \mathfrak{Y}, \mathfrak{Z},$

$\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, 1, 2, 3$

Matematyczne bezszeryfowe:  $\mathsf{A}$  \mathsf{A} ... itd.

$\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}, \mathsf{G}, \mathsf{H}, \mathsf{I}, \mathsf{J}, \mathsf{K}, \mathsf{L}, \mathsf{M}, \mathsf{N}, \mathsf{O}, \mathsf{P}, \mathsf{Q}, \mathsf{R}, \mathsf{S}, \mathsf{T}, \mathsf{U}, \mathsf{V}, \mathsf{W}, \mathsf{Y}, \mathsf{Z},$

$\mathsf{a}, \mathsf{b}, \mathsf{c}, 1, 2, 3$

Matematyczne pogrubione:  $\mathbf{A}$  \mathbf{A} ... itd.

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}, \mathbf{M}, \mathbf{N}, \mathbf{O}, \mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{S}, \mathbf{T}, \mathbf{U}, \mathbf{V},$

$\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{a}, \mathbf{b}, \mathbf{c}, 1, 2, 3$

## Niektóre ograniczniki

---

$\rangle$	<code>\rangle</code>	$\langle$	<code>\langle</code>	$\rfloor$	<code>\rfloor</code>
$\lfloor$	<code>\lfloor</code>	$\mid$	<code>\mid</code>	$\text{---}$	<code>\text{---}</code>
$\backslash$	<code>\backslash</code>	$/$	<code>/</code>		

## Różne inne przydatne symbole

---

$a \dots b$	<code>\dots</code>	$a \cdots b$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\partial$	<code>\partial</code>	$\ell$	<code>\ell</code>
$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>	$\angle$	<code>\angle</code>	$\square$	<code>\Box</code>

## ROZDZIAŁ 8

---

# Pakiet geometry

---

## Wstęp

---

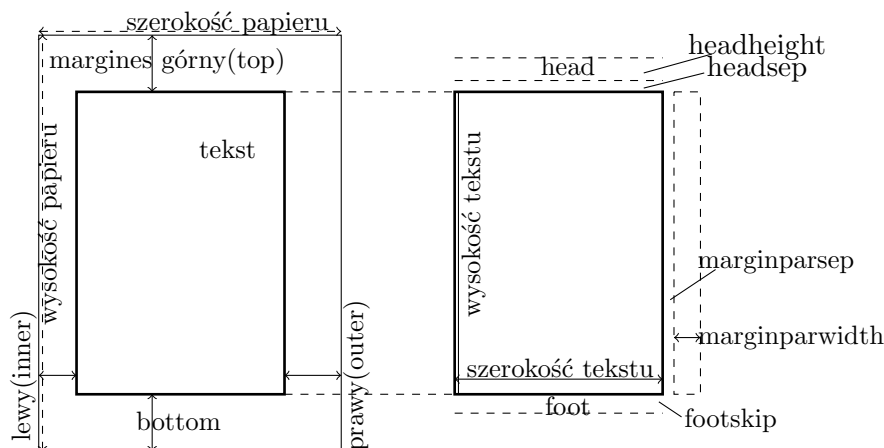
Pakiet `geometry` wczytuje się poleceniem `\usepackage{}`. Pakiet posiada bogaty zbiór dodatkowych parametrów, które omówimy w dalszej części tego dokumentu. Niektóre z nich są wczytywane wraz z pakietem z przypisanymi im wartościami domyślnymi. Jeżeli chcemy zmienić ich domyślne wartości parametrów odpowiadających za np. szerokość marginesów, możemy je umieścić jako parametry polecenia `\usepackage` lub wymienić je jako argument polecenia `\geometry{}`.

### Przykład 1.

```
\usepackage[layout=a5paper,landscape]{geometry}  
\geometry{showframe, hoffset=2cm}
```

Jeśli użyjemy opcji `includehead` i `includefoot`, wówczas cały dokument zostanie rozmieszczony w prostokącie o bokach „wysokość tekstu” i „szerokość tekstu”.

Strona tekstu składa się z trzech części (długości) w każdą stronę: jedno „total body” i dwa marginesy. Wystarczy zdefiniować dwie wielkości, trzecia zostanie wyliczona przez mechanizm auto-uzupełniania.



Rysunek 8.1: Organizacja strony

## Parametry strony w pakiecie geometry

---

Strona składa się z części zadrukowanej (total body) i marginesów. Na część zadrukowaną składają się: obszar tekstu (text area) z opcjonalnymi stopką, nagłówkiem i notkami na marginesie (marginpar). Marginesy są cztery: lewy, prawy, górny i dolny. Dla dokumentów dwustronnych poziome marginesy powinny nazywać się wewnętrzny i zewnętrzny.

## Polecenia pakietu geometry

---

W pakiecie zdefiniowane są następujące polecenia:

- `\geometry{<parametry>}`
- `\newgeometry{<parametry>}` i `\restoregeometry`
- `\savegeometry{<nazwa>}` i `\loadgeometry{<nazwa>}`

Polecenie `\geometry{<opcje>}` zmienia układ strony zgodnie z wartościami parametrów przywołanych w argumencie. Polecenie to powinno być

umieszczone w preambule, a więc przed `\begin{document}`. Jeśli polecenie `\geometry{<parametry>}` wpisujemy w preambule kilkakrotnie z różnymi parametrami, to wszystkie te parametry zostaną zastosowane łącznie.

Polecenie `\newgeometry{<parametry>}` zmienia ustawienia strony wewnątrz dokumentu, a polecenie `\restoregeometry` przywraca ustawienia poprzednie.

Polecenia `\savegeometry{<nazwa>}` i `\loadgeometry{<nazwa>}` pozwalają na zapamiętanie ustawień strony i zastosowanie ich w dowolnym momencie.

Listę parametrów (opcji) `\geometry`, `\newgeometry` i `\usepackage` wypisujemy w postaci `<nazwa>=<wartość>` oddzielonych przecinkami. Lista ta musi czynić zadość następującym warunkom:

- parametry można wypisać w kilku liniach, o ile nie oddzielimy ich linią pustą,
- dodatkowe odstępy pomiędzy kolejnymi wartościami są ignorowane,
- działanie poszczególnych parametrów nie zależy od miejsca położenia na liście.

## Przykład 2.

```
\usepackage[ a5paper , hmargin = { 3 cm,
                        .6 in } , height
                        = 10in ]{geometry}
```

jest równoważne

```
\usepackage[a5paper,height=10in,hmargin={3cm,0.6in}]{geometry}.
```

Równoważne są też

```
\usepackage{geometry}
```

```
\geometry{a5paper,height=10in,hmargin={3cm,0.6in}}
```

oraz

```
\usepackage[a5paper]{geometry}
```

```
\geometry{height=10in,hmargin={3cm,0.6in}}
```

## Typy parametrów

---

W pakiecie `geometry` dostępne są cztery typy parametrów.

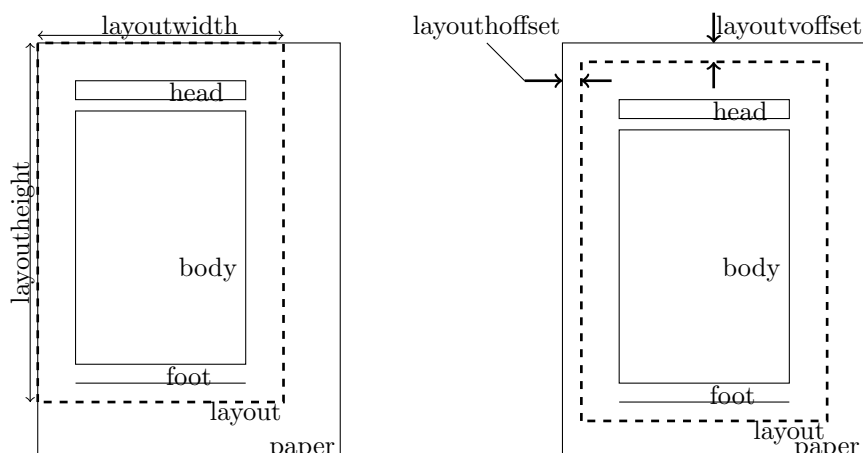
1. Przyjmujące wartości „prawda (true)”, „fałsz (false)” (boolean type)  
np. `twoside=true`. Jeśli nie ma podanej wartości, domyślną jest „prawda”. Nazwa (rozmiar) papieru jest wyjątkiem i powinien być podawany bez żadnej wartości.
2. Przyjmujące tylko jedną wartość np. `left=1.5cm, footskip=1cm`.
3. Przyjmujące dwie wartości `[nazwa]_{[wartość1],[wartość2]}`  
np. `hmargin={1cm,2cm}`
4. Przyjmujące trzy wartości `[nazwa]_{[wart1],[wart2],[wart3]}`. W każdym przypadku `[wartość]` musi oznaczać jakiś wymiar lub zero. Jako zero można zostawić puste miejsce lub wpisać znak „\*” np.:  
`hdivide={2cm,*,2cm}, vdivide={1cm,14cm, }`

## Kompozycja strony – dostępne parametry

---

Różnicę pomiędzy rozmiarem papieru i układem strony (layout) najlepiej pokazuje rysunek 8.2.

<code>layout</code>	–	ustala rozmiar strony według rozmiaru papieru. <code>layout=&lt;nazwa papieru&gt;</code> .
<code>layoutwidth</code>	–	szerokość strony.
<code>layoutheight</code>	–	wysokość strony.
<code>layoutsize</code>	–	rozmiar strony. <code>layoutsize={&lt;szerokość&gt;,&lt;wysokość&gt;}</code> lub <code>layoutsize={&lt;wielkość&gt;}</code>
<code>layouthoffset</code>	–	poziome przesunięcie względem lewego brzegu.
<code>layoutvoffset</code>	–	pionowe przesunięcie względem górnego brzegu.
<code>layoutoffset</code>	–	oba poziome i pionowe przesunięcie.



Rysunek 8.2: Zależności pomiędzy rozmiarem papieru a układem strony.

## Rozmiar tekstu (body size)

---

Podamy teraz najczęściej używane opcje dla określenia wymiarów tekstu na stronie.

- |                   |   |
|-------------------|---|
| <b>hscale</b>     | – stosunek szerokości części zadrukowanej (total body) do szerokości papieru <code>\paperwidth</code> . <code>hscale=0.9</code> jest równoważna <code>width=0.9\paperwidth</code> . Domyślna wartość tego parametru to 0.7. |
| <b>vscale</b>     | – to samo co poprzednio ale w odniesieniu do wysokości.   |
| <b>scale</b>      | – stosunek szerokości części zadrukowanej (total body) do papieru.  |
| <b>width</b>      | – szerokość części zadrukowanej (total body).   |
| <b>height</b>     | – wysokość części zadrukowanej (total body).  |
| <b>total</b>      | – szerokość i wysokość części zadrukowanej (total body).  |
| <b>textwidth</b>  | – ustala szerokość tekstu.  |
| <b>textheight</b> | – ustala wysokość tekstu.   |
| <b>text body</b>  | – ustala szerokość i wysokość tekstu.<br><code>body={&lt;szerokość&gt;, &lt;wysokość&gt;}</code><br>lub <code>text=&lt;wartość&gt;</code> .   |



- `lines` – pozwala użytkownikowi ustalić wysokość (`\textheight`) tekstu poprzez podanie ilości linii tekstu na stronie `lines=<liczba>`.
- `includehead` – wysokość nagłówka wraz z odstępem od tekstu wliczana jest do wysokości części zadrukowanej (total body)

## Marginesy

---

Parametry wymienione poniżej pozwalają ustalić wielkości marginesów w dokumencie.

- `left|lmargin|inner` – margines lewy (druk jednostronny) lub wewnętrzny (druk dwustronny). Jest to odległość od lewej krawędzi papieru do lewego brzegu total body (określenia wyjaśnione w tabeli ??).
- `right|rmargin|outer` – prawy lub zewnętrzny (druk dwustronny) margines, `\right=<szerokość>`.
- `top|tmargin` – margines górny na stronie `\top=<szerokość>`. Ta opcja nie ma nic wspólnego z poleceniem `\topmargin`.
- `bottom|bmargin` – margines dolny na stronie.
- `hmargin` – lewy i prawy margines ustalany jednocześnie poleceniami `hmargin={<lewy>,<prawy>}` lub `hmargin=<wielkość>`.
- `vmargin` – górny i dolny margines ustalany jednocześnie poleceniami `vmargin={<górn>,<dół>}` lub `vmargin=<wielkość>`.
- `hmarginratio` – stosunek wielkości lewego i prawego marginesu. Stosunek ten powinien być zapisany liczbami całkowitymi ;100, oddzielonymi dwukropkiem np. 2:3. Domyślnie jest 1:1 dla druku jednostronnego i 2:3 dla druku dwustronnego.

<code>vmarginratio</code>	– stosunek wielkości marginesów dolnego i górnego. Domyślnie 2:3.
<code>centering</code>	– automatycznie centruje obszar tekstu na stronie i jest równoważny z <code>marginratio=1:1</code> . Domyślna wartość jest „false”.
<code>bindingoffset</code>	– rezerwuje na oprawę część obszaru z lewej strony w druku jednostronnym, a od strony wewnętrznej w druku dwustronnym. <code>bindingoffset=&lt;wielkość&gt;</code> .

Wszystkie te opcje zmieniają bądź modyfikują wartości domyślne parametrów strony wczytywane razem z klasą dokumentu.

### Inne parametry opcjonalne

---

<code>mag</code>	– ustala wartość powiększenia ( <code>\mag</code> ) i automatycznie modyfikuje wielkości <code>\hoffset</code> i <code>\voffset</code> . Podaje się go w postaci <code>mag=&lt;liczba&gt;</code> . Wartość „liczba” powinna być liczbą całkowitą. Liczba 1000 oznacza rozmiar normalny (dla danego papieru). Dla <code>a4paper</code> <code>mag=1414</code> będzie oznaczało powiększenie do rozmiaru <code>a3paper</code> , który jest $1.414(= \sqrt{2})$ razy większy niż <code>a4paper</code> . W tym przypadku powiększone zostaną także wszystkie wielkości podane jako „true”, na przykład <code>1.5truein</code> .
<code>truedimen</code>	– zmienia wszystkie wewnętrzne wielkości na takie, które nie będą podlegały zwiększeniu gdy powiększymy dokument poleceniem <code>\mag</code> . Zazwyczaj ten parametr jest używany z poleceniem <code>\mag</code> .
<code>showframe</code>	– rysuje ramki wokół wszystkich zdefiniowanych obszarów strony takich jak tekst, nagłówki, stopka itd.

`showcrop` – drukuje znaki we wszystkich rogach zdefiniowanego przez użytkownika układu strony.

## Ustawienia domyślne

---

Domyślne ustawienia parametrów strony są następujące:

1. Marginesy górny i dolny są w stosunku 2/3.
2. Marginesy poziome zależą od tego, czy dokument jest jedno czy dwustronny i tak:

lewy(wewnętrzny) : prawy(zewnętrzny) =

$$\begin{cases} 1 : 1 & \text{dla dokumentu jednostronnego} \\ 2 : 3 & \text{dla dokumentu dwustronnego} \end{cases}$$

Dla dokumentu jednostronnego pakiet `geometry` ma następujące domyślne parametry strony:

1. `scale=0.7` („body” to  $0.7 \times \text{paper}$ ),
2. `marginratio= \{1 : 1, 2 : 3\}` (1:1 – marginesy poziome, 2:3 – marginesy pionowe),
3. `ignoreall` (nagłówek, stopka i notki na marginesie nie są brane pod uwagę, gdy wyliczane są wymiary „body”).

Dla dokumentu dwustronnego z parametrem `twoside`, domyślne ustawienia są takie same jak dla jednostronnego z wyjątkiem marginesów poziomych, które są ustalone w stosunku 2:3.

## ROZDZIAŁ 9

---

# Prezentacje w klasie beamer

---

## Wstęp

---

Klasa BEAMER jest klasą L<sup>A</sup>T<sub>E</sub>X-a do tworzenia prezentacji które przygotowuje się inaczej niż w typowych programach WYSIWYG. Autorzy to Till Tantau, Joseph Wright, Vedran Miletic. Nazwa programu pochodzi od niemieckiego słowa *Beamer*, pseudoanglicyzmu, oznaczającego video projektor. Dokument złożony w klasie BEAMER posiada strukturę dokumentu L<sup>A</sup>T<sub>E</sub>X-a, a więc preambułę i dokument właściwy, w którym umieszcza się między innymi rozdziały, punkty i ramki (frames). Aby korzystać z BEAMER -a trzeba znać L<sup>A</sup>T<sub>E</sub>X-a.

Instalacja programu jest prosta, można użyć standardowego instalatora i skorzystać z opcji „dodaj pakiet”. W większości dystrybucji L<sup>A</sup>T<sub>E</sub>X-owych wszystkie pliki BEAMER -a instalowane są automatycznie. Najnowszą wersję pakietów można znaleźć pod adresem <http://sourceforge.net/projects/latex-beamer/>. Klasa BEAMER jest udostępniana na licencji GNU, wersja 2.

Klasa BEAMER ma wiele mechanizmów które przesądzają o jej przydatności do tworzenia profesjonalnych prezentacji. Najważniejsze z nich to:

1. Można używać wszystkich komend L<sup>A</sup>T<sub>E</sub>X-a.
2. Łatwość umieszczania w prezentacji efektów animacji i elementów dynamicznych.
3. Szeroki wachlarz predefiniowanych szablonów prezentacji (thems).

4. Narzędzia do definiowania własnych szablonów lub przeddefiniowania już istniejących.
5. Specjalny styl, który umożliwia umieszczenie w jednym dokumencie  $\text{\LaTeX}$ -owym treści wykładu i przygotowanych do wyświetlenia slajdów prezentacji.
6. Możliwość tworzenia prezentacji multimedialnych.

Klasa `BEAMER` wprowadza własny dialekt  $\text{\LaTeX}$ -a do tworzenia slajdów. Slajdy mogą być pokazywane na ekranie etapami odkrywając fragmenty tekstu, które były dotąd schowane lub zakryte. Umożliwia tworzenie „handouts” tj. materiałów drukowanych do rozdawania słuchaczom lub do opublikowania w sieci. Opcja *article* służy do wydrukowania treści prezentacji na standardowych arkuszach A4.

## Dokument

---

### Struktura

---

Prezentację przygotowujemy według następującego schematu

```
\documentclass{beamer}
\usepackage{nazwy pakietów}
\mode<presentation>
\title{Tytuł}
\author{Nazwisko autora}
\date{data}
\begin{document}
\begin{frame}
\titlepage
\end{frame}
\begin{frame}
\frametitle{Outline}
\tableofcontents
```

```

\end{frame}
\section{Tytuł}
\subsection{Tytuł}
\begin{frame}
\frametitle{Tytuł slajdu}
treść
.....
\end{frame}
\end{document}

```

Prezentacja składa się z szeregu ramek (frame) z których każda może składać się z kilku slajdów tworzących kolejno wyświetlane warstwy (overlays). Możemy w ten sposób uzyskać efekt animacji tekstu.

### Przykład 3.

```

\begin{frame}{Przykładowy slajd}{Tworzenie}
Treść slajdu
\end{frame}

```

### Przykład 4.

```

\frame{\frametitle{Przykładowy slajd}
\framesubtitle{Tworzenie}
Treść slajdu
}

```

## Slajd „zwykły” – tytuł i podtytuł slajdu

---

Z ogólnych zasad przygotowywania prezentacji wynika, że każdy slajd powinien mieć tytuł. Szablon BEAMER-a umieszcza tytuł slajdu za pomocą polecenia: `\frametitle{Tytuł slajdu}`.

Podtytuł dodajemy poleceniem `\framesubtitle{Podtytuł slajdu}`.

## Slajd przepełniony

---

Jeżeli umieścimy pomiędzy `\begin{frame}` i `\end{frame}` zbyt dużo tekstu to nie zostanie on automatycznie przeniesiony do następnej ramki tylko wyświetlony poza nią. Co wtedy możemy zrobić?

- „Ręcznie” podzielić slajd na mniejsze części.
- Mamy też możliwość ustawienia automatycznego dzielenia treści na poszczególne ramki. Korzysta się z niej najczęściej przy umieszczaniu na slajdach długich spisów, np. literatury. Należy wówczas użyć opcji `allowframebreaks`.

### Przykład 5.

```
\begin{frame}[allowframebreaks]{Spis literatury}
\begin{thebibliography}{99}
\bibitem
\bibitem
...
\end{thebibliography}
\end{frame}
```

- Zmniejszyć rozmiar fontu `\documentclass[9pt]{beamer}`
- `\begin{frame}[shrink=N]{Tytuł slajdu}`  
Treść slajdu  
`\end{frame}`

## Slajd tytułowy

---

Do wygenerowania slajdu tytułowego używamy następującej konstrukcji:

```
\documentclass{beamer}
\usepackage{}
. . .
```

```

\title{Tytuł prezentacji}
\subtitle{Podtytuł prezentacji}
\author{Autor1\inst{1} \and Autor2 \inst{2}}
\institute{\inst{1} Afiliacja1 \and \inst{2} Afiliacja2}
\date{Łódź, \today}
\titlegraphic{\pgfimage[height=1.5 cm]{logo-1}}
\logo{\pgfimage[height=0.5 cm]{logo-2}}
\begin{document}
\maketitle % równoważne \frame{titlepage}
. . .
\end{document}

```

## Warstwy

---

- Domyślnie slajd ma jedną wersję.
- Warstwy – kolejno wyświetlane wersje slajdu.
- Warstwy są dodawane kiedy zastosujemy instrukcje:
  - \pause
  - \onslide
  - \only
  - \uncover
  - \visible
  - specyfikatora warstw postaci  $\langle \dots \rangle$ .

**Przykład 6.** Przykłady specyfikatorów:

- $\langle 2 - 4 \rangle$  dotyczy warstw od 2 do 4,
- $\langle -3 \rangle$  dotyczy warstw od 1 do 3,
- $\langle 3 - \rangle$  dotyczy warstw od 3 do ostatniej,
- $\langle 3, 5 \rangle$  dotyczy warstwy 3 i warstwy 5
- $\langle 1 - 3, 6 - \rangle$  dotyczy warstwy od 1 do 3 oraz od 6 do ostatniej.



## Instrukcje sterujące warstwami

---

Komendy	Opis
<code>\only&lt; ... &gt;\{Tekst\}</code>	Wstaw Tekst w podanych warstwach
<code>\uncover&lt; ... &gt;\{Tekst\}</code>	Wstaw Tekst w podanych warstwach oraz zarezerwuj miejsce we wszystkich warstwach
<code>\visible&lt; ... &gt;\{Tekst\}</code>	Tak jak <code>\uncover</code> , ale nie uwzględniaj efektu przezroczystości

### Instrukcja `\only`

#### Przykład 7. Specyfikatory `< 1 >`, `< 2 >`.

```
\only<1>{
\begin{block}{}
Po pierwsze
\end{block}}
\only<2>{
\begin{block}{}
Po drugie
\end{block}}
```

Ramka zdefiniowana w ten sposób będzie składała się z dwóch slajdów, których treść będzie kolejno zastępowana treścią następnego.

#### Przykład 8. Specyfikatory `< 1- >`, `< 2- >`.

```
\only<1->{
\begin{block}{}
Po pierwsze
\end{block}}
\only<2->{
```

```
\begin{block}{}

```

Po drugie

```
\end{block}}
```

Ramka zdefiniowana w ten sposób będzie składała się z dwóch slajdów, których treść będzie kolejno dopisywana.

**Instrukcja** `\uncover`

**Przykład 9. Specyfikatory** `< 1 >`, `< 2 >`.

```
\uncover<1>{

```

```
\begin{block}{}

```

Po pierwsze

```
\end{block} }
```

```
\uncover<2>{

```

```
\begin{block}{}

```

Po drugie

```
\end{block} }
```

Instrukcja `\uncover` ma za zadanie wyświetlenie kolejno grup tekstu w takim miejscu slajdu w jakim zostały one umieszczone, to znaczy tekst choć tekst nie jest cały czas widoczny, ale miejsce dla niego jest zarezerwowane.

**Przykład 10. Specyfikatory** `< 1- >`, `< 2- >`.

```
\uncover<1->{

```

```
\begin{block}{}

```

Po pierwsze

```
\end{block} }
```

```
\uncover<2->{

```

```
\begin{block}{}

```

Po drugie

```
\end{block} }
```

**Instrukcja** `\visible`

**Przykład 11. Specyfikatory**  $\langle 1 \rangle$ ,  $\langle 2 \rangle$ .

```
\visible<1>{
\begin{block}{}
Po pierwsze
\end{block}}
\visible<2>{
\begin{block}{}
Po drugie
\end{block}}
```

**Przykład 12. Specyfikatory**  $\langle 1- \rangle$ ,  $\langle 2- \rangle$ .

```
\visible<1->{
\begin{block}{}
Po pierwsze
\end{block} }
\visible<2->{
\begin{block}{}
Po drugie
\end{block} }
```

**Instrukcja** `\pause`

**Przykład 13.**

```
\begin{description}
\setlength{\itemsep}{0pt}
\item[Koło] zbiór wszystkich punktów płaszczyzny,
których odległość od ustalonego punktu na tej
płaszczyźnie (środku koła) nie przekracza pewnej
```

```
wartości (promienia koła). \pause
\item[Prosta] jedno z podstawowych pojęć geometrii,
szczególny przypadek nieograniczonej z obydwu stron
krzywej o nieskończonym promieniu krzywizny w każdym
punkcie.
\end{description}
```

**Przykład 14.** Pojawiające się elementy w liście numerowanej,  
specyfikatory `< 1- >`, `< 2- >`.

```
\begin{enumerate}
\item<1-> Punkt pierwszy, widoczny we wszystkich
warstwach.
\item<2-> Punkt drugi, widoczny począwszy od
warstwy drugiej.
\item<2-> Punkt trzeci, również widoczny
począwszy od warstwy drugiej.
\item<3-> Punkt czwarty, widoczny począwszy od
warstwy trzeciej.
\end{enumerate}
```

**Przykład 15.** Kolejno pojawiające się elementy,  
specyfikator `< +- >`.

```
\begin{enumerate}[<+->]
\item Punkt pierwszy, widoczny we wszystkich
warstwach.
\item Punkt drugi, widoczny począwszy od
warstwy drugiej.
\item Punkt trzeci, widoczny począwszy od
warstwy trzeciej.
\item Punkt czwarty, widoczny począwszy od
warstwy czwartej.
```

```
\end{enumerate}
```

**Przykład 16.** Wyróżnianie kolejnych elementów w liście nienumerowanej, specyfikator `< 3 – |alert@3 >`.

```
\begin{itemize}
\item<3- | alert@3> Punkt pierwszy, widoczny
począwszy od warstwy trzeciej
\item<1- | alert@1> Punkt drugi, widoczny we
wszystkich warstwach
\item<2- | alert@2> Punkt trzeci, widoczny
począwszy od warstwy drugiej
\end{itemize}
```

**Przykład 17.** Wyróżnianie kolejno pojawiających się elementów w liście nienumerowanej, specyfikator `< + – |alert@+ >`.

```
\begin{itemize}[<+- | alert@+>]
\item Punkt pierwszy, widoczny we wszystkich
warstwach.
\item Punkt drugi, widoczny począwszy od
warstwy drugiej.
\item Punkt trzeci, widoczny począwszy od
warstwy trzeciej.
\end{itemize}
```

**Przykład 18.** Elementy pojawiające się nie po kolei.

```
\begin{itemize}
\item Punkt pierwszy, widoczny począwszy od
warstwy pierwszej.
\onslide<3>
\item Punkt drugi, widoczny w warstwie trzeciej.
```

```

\onslide<2-3>
\item Punkt trzeci, widoczny w warstwie drugiej i
trzeciej.
\onslide
\item Punkt czwarty, widoczny we wszystkich
warstwach.
\end{itemize}

```

## Rodzaje wyróżnień

Mechanizm tworzenia warstw możemy także wykorzystać do zaznaczania innym kolorem kolejnych informacji na wyświetlanym slajdzie.

```

\begin{frame}
\textcolor<1>{red}{Ten tekst pierwszy będzie czerwony.}
\textcolor<2>{red}{Ten tekst będzie czerwony jako drugi.}
\textcolor<3>{red}{A ten stanie się czerwony jako ostatni.}
\end{frame}

```

Ramka zdefiniowana jak wyżej będzie składała się z trzech slajdów. Na pierwszym zobaczymy pierwsze zdanie w kolorze czerwonym, a pozostałe będą czarne. Na drugim czerwone będzie tylko zdanie drugie, a pozostałe będą czarne, a na trzecim czerwone będzie tylko ostatnie zdanie. Jeżeli pomiemy w pierwszej linii parametr <1> to wówczas pierwsze zdanie pozostanie czerwone na wszystkich slajdach.

Polecenie zmiany koloru można zastąpić dowolnym innym, np. poleceniem zmiany kroju (`\textit`, `\textsc`) lub grubości (`\textbf`).

## Stopniowe odsłanianie wzoru matematycznego

Najwygodniej w przypadku wzoru matematycznego jest skorzystać z instrukcji `\pause` jak w przykładzie poniżej.

```

\begin{displaymath}
a \ \text{\pause} = \text{\pause} \ \frac{x+y}{\text{\pause}} \ \text{\pause} \ y-z

```

```
\end{displaymath}
```

### Odślanianie tabeli wierszami

```
\rowcolors{2}{gray!20}{gray!30}
\begin{tabular}{l|cccc}
Klasa & A & B & C & D \\ \hline
X & 1 & 2 & 3 & 4 \pause \\
Y & 3 & 4 & 5 & 6 \pause \\
Z & 5 & 6 & 7 & 8 \\
\end{tabular}
```

### Odślanianie tabeli kolumnami

```
\rowcolors{2}{gray!20}{gray!30}
\begin{tabular}{l|c<\onslide<2->>c<{\onslide<3->>c<{\onslide<4->>c<{\onslide}}
Klasa & A & B & C & D \\ \hline
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8 \\
\end{tabular}
```

### Grupowanie tekstu w bloki

---

BEAMER dostarcza nam prostego narzędzia do grupowania zawartości slajdu w bloki. Każdy blok może (ale nie musi) posiadać tytuł, a jego wygląd zależy od wybranego motywu prezentacji.

<code>\begin{block}{Tytuł bloku}</code>	<code>\begin{block}{}{}</code>
Zawartość bloku	Zawartość bloku
<code>\end{block}</code>	<code>\end{block}</code>

Mamy także do dyspozycji inne już zdefiniowane bloki, między innymi `alertblock` i `exampleblock`. W pierwszym zawartość jest zaznaczana

innym kolorem (najczęściej czerwonym), drugi natomiast jest używany do przytaczania przykładów.

## Podział slajdu na kolumny

---

Zawartość slajdu możemy także podzielić na kolumny. Służy do tego środowisko `columns`. Przykład użycia tego środowiska czytelnik znajdzie poniżej.

```
\begin{columns}[t]
% Pierwsza kolumna
\begin{column}{0.5\textwidth}
Dwie \ \ linie.
\end{column}
% Druga kolumna
\begin{column}{0.5\textwidth}
Jedna linia (ale wyrównana).
\end{column}
\end{columns}
```

```
\begin{columns}[t]
\column{0.5\textwidth} %Pierwsza kolumna
Dwie \ \ linie.
\column{0.5\textwidth} %Druga kolumna
Jedna linia (ale wyrównana).
\end{columns}
```

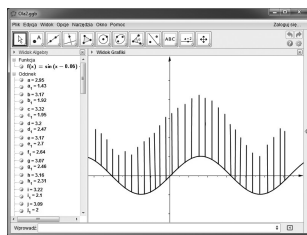
## Grafika w prezentacji

---

Do prezentacji wstawiamy rysunki tak samo jak do pliku tekstowego czyli korzystając z możliwości pakietu `GRAPHICX`.

```
\includegraphics[width=4cm]{rysunek1.jpg}
```





## Motywy prezentacji

---

Użytkownik BEAMER -a ma do dyspozycji szeroką gamę różnorodnych motywów prezentacji.

Motyw\_prezentacji = Motyw\_kolorystyczny + Motyw\_czcionki + Motyw\_wewnętrzny + Motyw\_zewnętrzny

Motywy różnią się nie tylko kolorami ale także sposobem organizacji informacji na slajdzie. I tak na przykład:

- Bez pasków nawigacyjnych na slajdzie – motywy
  - default, boxes, Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS, Pittsburgh, Rochester
- Z drzewem sekcji i podsekcji na slajdzie – motywy
  - Antibes, JuanLesPins, Montpellier
- Z paskiem bocznym zawierającym spis treści – motywy
  - Berkeley, PaloAlto, Goettingen, Marburg, Hannover
- Z mini ramką nawigacyjną – motyw
  - Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt, Singapore, Szeged
- Z tabelami sekcji i podsekcji
  - Copenhagen, Luebeck, Malmoe, Warsaw

## Dodatek

---

### Hiperłącza i przyciski

---

Skok do następnego slajdu realizuje zestaw instrukcji opisany poniżej.

```
\begin{theorem}
. . .
\end{theorem}
\onslide<1>
\hfill\hyperlinkframestartnext{
\beamerskipbutton{Pomiń dowód twierdzenia}}
\onslide<2>
\begin{proof}
. . .
\end{proof}
```

### Powiększanie fragmentów slajdu

---

```
\begin{frame}
\frametitle<1>{Mapa odsyłaczy}
\framesubtitle<1>{Powiększanie fragmentów slajdu}
\framezoom<1><2>[border](1.5cm,0.25cm)(2.5cm,2.5cm)
\framezoom<1><3>[border](2cm,3.5cm)(2cm,1.5cm)
\pgfimage[height=7cm]{lasiczka.jpg}
\end{frame}
```

### Elementy multimedialne

---

Filmy i klipy dźwiękowe wstawiamy do prezentacji po uprzednim wczytaniu pakietu `multimedia` i odtwarzamy poleceniem `\movie` – patrz poniżej.

```
\usepackage{multimedia}
. . .
\begin{document}
```

```

. . .
\movie[label=film,width=4cm,height=3cm,poster,showcontrols,
duration=4s]{}{xxx.avi}
\hyperlinkmovie[start=1s,duration=2s]{film}{
\beamerbutton{Odtwórz środkową część filmu}
}
\hyperlinkmovie[start=3s,duration=1s]{film}{
\beamerbutton{Odtwórz końcową część filmu}
}
. . .
\end{document}

```

Pakiet `multimedia` umożliwia też odtwarzanie klipów dźwiękowych. Należy wtedy użyć polecenie `\sound`.

```

\usepackage{multimedia}
. . .
\sound[autostart]{zzz.wav}

```

## Animowanie tekstów

Możemy także animować tekst na slajdzie – przykład poniżej.

```

\newdimen\offset
\begin{frame}
\animate<2-14>
\animatevalue<1-15>{\offsetg}{-5cm}{0cm}
\hskip\offset
\begin{minipage}{\textwidth}
\begin{theorem}
Wjeżdżające twierdzenie
\end{theorem}
\end{minipage}
\end{frame}

```

# Bibliografia

---

- [1] Robert Bringhurst, *Elementarz stylu w typografii*, Design Plus, Kraków 2007.
- [2] Robert Chwałowski, *Typografia typowej książki*, Helion, 2002.
- [3] Michael Goossens, Frank Mittelbach, Alexander Samarin, *The L<sup>A</sup>T<sub>E</sub>X Companion – Second Edition*, Addison-Wesley, 2005.
- [4] Andrzej Icha, *L<sup>A</sup>T<sub>E</sub>X2e dla matematyków*, Wydawnictwo Akademii Pomorskiej w Słupsku, Słupsk 2007.
- [5] Donald E. Knuth, *T<sub>E</sub>X. Przewodnik użytkownika*, WNT, Warszawa 2005.
- [6] Helmut Kopka, Patrick W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X2e*, Addison-Wesley, second edition 1997.
- [7] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X. Podręcznik i przewodnik użytkownika*, WNT, Warszawa 2004.
- [8] Wojciech Myszka, Ewaryst Rafajłowicz, *L<sup>A</sup>T<sub>E</sub>X zaawansowane narzędzia*, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1996.
- [9] Piet van Oostrum, *Page layout in L<sup>A</sup>T<sub>E</sub>X*, [www.ctan.org](http://www.ctan.org)
- [10] Tomasz Przechlewski, *Praca magisterska i dyplomowa z programem L<sup>A</sup>T<sub>E</sub>X*, Wolters Kluwer Polska Sp. z o.o, 2011.

- [11] Till Tantau, Joseph Wright, Vedran Miletić, <http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
- [12] Hideo Umei, *The Geometry Package*, [www.ctan.org/pkg/geometry](http://www.ctan.org/pkg/geometry)
- [13] Herbert Voß, *Math mode – v. 2.47*, [www.tex.ac.uk/tex-archive/info/math/.../mathmode/Mathmode.pdf](http://www.tex.ac.uk/tex-archive/info/math/.../mathmode/Mathmode.pdf)
- [14] Zofia Walczak, *L<sup>A</sup>T<sub>E</sub>X dla niecierpliwych – część I*, wydanie drugie, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2014.

NOWOCZESNY  
NAUCZYCIEL  
MATEMATYKI



publikacja bezpłatna



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



WYDAWNICTWO  
UNIwersytetu  
ŁÓDZKIEGO

[www.wydawnictwo.uni.lodz.pl](http://www.wydawnictwo.uni.lodz.pl)  
e-mail: [ksiegarnia@uni.lodz.pl](mailto:ksiegarnia@uni.lodz.pl)  
tel. (42) 665 58 63, faks (42) 665 58 62

